



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

1976

Static and dynamic buckling of shallow spherical shells subjected to a axisymmetric and nearly axisymmetric step pressure loads using SATANS-IIA, a modified version of SATANS-II.

Shutt, Michael D.

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/17952>

---

*Downloaded from NPS Archive: Calhoun*



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECTED TO AXISYMMETRIC AND NEARLY AXISYMMETRIC STEP PRESSURE LOADS USING SATANS-IIA, A MODIFIED VERSION OF SATANS-II

Michael D. Shutt

KNOX LIBRARY,  
POSTGRADUATE SCHOOL  
REY, CALIF. 93940

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL  
SHELLS SUBJECTED TO AXISYMMETRIC AND NEARLY  
AXISYMMETRIC STEP-PRESSURE LOADS USING SATANS-IIA,  
A MODIFIED VERSION OF SATANS-II

by

Michael D. Shutt

December 1976

Thesis Advisor:

Robert E. Ball

Approved for public release; distribution unlimited.

T 17665



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Static and Dynamic Buckling of Shallow Spherical Shells Subjected To Axisymmetric and Nearly Axisymmetric Step-Pressure Loads Using SATANS-IIA A Modified Version of SATANS-II		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1976
7. AUTHOR(s) Michael D. Shutt		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1976
		13. NUMBER OF PAGES 167
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Buckling Shallow Spherical Shells Axisymmetric Asymmetric Computer Program		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and nearly axisymmetric step-pressure loads of infinite duration. A comparison was made between the new buckling results and previous results obtained without the new pole routine. The comparison revealed a significant		



20. (continued)

change in the buckling pressures, due solely to the change in the pole routine. The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells.



STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS  
SUBJECTED TO AXISYMMETRIC AND NEARLY AXISYMMETRIC STEP  
PRESSURE LOADS USING SATANS-IIA, A MODIFIED VERSION OF  
SATANS-II

by

Michael D. Shutt  
Lieutenant  
B.S., Oregon State University, 1970

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the  
NAVAL POSTGRADUATE SCHOOL  
December 1976



## ABSTRACT

A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and dynamic nearly axisymmetric step-pressure loads of infinite duration. A comparison was made between the new buckling results and previous results obtained without the new pole routine. The comparison revealed a significant change in the buckling pressures, due solely to the change in the pole routine. The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells.



## TABLE OF CONTENTS

I. INTRODUCTION.....	9
II. DESCRIPTION OF SATANS-IIA.....	12
III. IMPROVED POLE ROUTINE.....	15
IV. PROBLEM DESCRIPTION.....	20
V. RESULTS AND DISCUSSION.....	24
A. STATIC AXISYMMETRIC BUCKLING ANALYSIS.....	24
B. DYNAMIC AXISYMMETRIC BUCKLING ANALYSIS.....	24
C. DYNAMIC ASYMMETRIC BUCKLING ANALYSIS.....	25
VI. SUMMARY AND CONCLUSIONS.....	30
Appendix A: LISTING OF SATANS-IIA.....	44
Appendix B: LISTING OF OUTPUT FROM EXAMPLE PROBLEM.....	142
Appendix C: INPUT DATA GUIDE FOR SATANS-IIA.....	150
Appendix D: LISTING OF NEW POLE ROUTINE FOR SATANS-IIA.....	157
Appendix E: LISTING OF CARDS FOR $\bar{V}$ AND $\bar{V}_{MAX}$ .....	161
LIST OF FIGURES AND TABLES.....	163
LIST OF REFERENCES.....	165
INITIAL DISTRIBUTION LIST.....	167



# LIST OF SYMBOLS

$b$	= nondimensional inplane stiffness
$E$	= the modulus of elasticity of the shell
$H$	= the rise of the spherical cap at the pole
$h$	= the thickness of the shell
$m$	= the mass density of the shell
$M_s$	= the meridional bending moment per unit length
$n$	= the Fourier index
$P$	= a nondimensional applied load
$P_{CRIT}$	= the nondimensional critical pressure
$q_o$	= the classical buckling pressure of a complete sphere
$q^{(n)}$	= a column matrix containing the coefficients of the $n^{th}$ term in the series expansion of the applied load
$r$	= the normal distance from the axis of revolution to the surface of the cap
$r_o$	= the normal distance from the axis to the cap in the base plane; the maximum value of $r$
$R_s, R_\theta$	= the radii of curvature in the $s$ and $\theta$ directions, respectively
$s$	= the meridional distance along the surface of the shell
$t$	= the nondimensional time
$T$	= the time
$T_o$	= a reference time



$U, V, W$  = the displacements in the  $s$ ,  $\theta$  and  $\mathcal{J}$  directions, respectively  
 $u, v, w$  = nondimensional series coefficients of  $U, V, W$   
 $\bar{V}$  = a nondimensional measure of the volume of the shell deformation  
 $\bar{V}_{MAX}$  = the peak in the time history of the parameter  $\bar{V}$   
 $w^{(n)}$  = the displacement in the  $\mathcal{J}$  direction in the  $n^{th}$  harmonic  
 $\delta t$  = the nondimensional time increment  
           = distance between stations  
 $\epsilon^{(n)}$  = the nondimensional parameter governing the magnitude of the load applied in the asymmetric harmonics  
 $\mathcal{J}$  = the coordinate normal to the surface of the shell  
 $\theta$  = the circumferential angle measured about the axis of revolution  
 $\lambda$  = a nondimensional geometric parameter used to describe the spherical cap  
 $\nu$  = Poisson's ratio  
 $\xi$  = the normal distance from the base plane to the middle surface of the undeformed cap



## ACKNOWLEDGEMENT

I wish to acknowledge the help that I have received in the course of my thesis research.

I am particularly indebted to Professor Robert E. Ball, who provided me with the guidance and aid necessary to complete my project.

I am also grateful to Professor Johann Arbocz of CALTECH, who provided the initial deck of cards from which I worked.

I also wish to give credit to the NPS programming consultants, and in particular Richard Donat, for the expert assistance he provided in debugging my changes to the computer program.

I especially want to express my sincere appreciation to my wife, Roberta, for her patience and assistance in typing of this thesis.



## I. INTRODUCTION

In 1973 a digital computer study was presented by Ball and Burt [1] for the dynamic buckling load of clamped shallow spherical shells subjected to axisymmetric and nearly axisymmetric step-pressure loads. A static buckling analysis of the same spherical shells had been carried out in 1970 by Stilwell and Ball [2]. In these two studies the digital computer program SATANS-I [3] was used to calculate the critical buckling pressures for a large range of shell sizes. Other studies of the buckling of shallow shells have been conducted by Huang [4,5], by Stephens and Fulton [6], by Lock et al. [7], by Stricklin [8], and most recently by Akkas [9]. In Reference 1 the results from these other studies, except for those by Akkas, are compared with the results from SATANS-I for both static and dynamic buckling. In the axisymmetric static analysis the comparison with the results obtained by Huang [4] revealed that the SATANS-I results were higher than Huang's results for several shell sizes. In the dynamic, axisymmetric buckling analysis the SATANS-I results again either agreed closely with, or were somewhat higher than, the results by Huang [5], Stephens and Fulton [6], and Stricklin [8]. However, it was noted then that there was a general lack of consistent agreement among any of the sets of results. As a consequence, it appeared at that time that the axisymmetric buckling problem had not yet been totally resolved and that additional studies would be appropriate.

In the asymmetric dynamic buckling analysis of Reference 1 the few comparisons that could be made for the critical load also indicated that the SATANS-I results may be too



high. A comparison of the recent estimates for the asymmetric dynamic buckling load obtained by Akkas [9] with the SATANS-I results also reveals the SATANS-I results to be well above those of Akkas [9]. However, it should be noted that the results obtained by Akkas were from his attempt to obtain a lower bound on the critical asymmetric load. This bound on the buckling load is obtained without the execution of a complete transient response analysis on the asymmetric part of the response of the shell, as is done in SATANS-I. In Akkas' analysis (Problem 1) the transient nonlinear axisymmetric response is computed, and a determinant is examined for possible bifurcation into asymmetric motion at each time step. The minimum load at which the determinant becomes zero is defined as the lower bound of the critical load.

As a consequence of the generally high buckling loads predicted by SATANS-I, a re-examination of the static and dynamic buckling of the shallow spherical shell was made in an attempt to determine the possible cause, or causes, of the high buckling loads. In our search we discovered that a modification of the manner in which the pole conditions are numerically approximated significantly lowered the buckling loads to values that are now in good agreement with the other results. The new procedure for handling the pole condition is given in section III of this thesis. The new buckling results are given in section V.

In addition to the pole condition modifications and the new buckling results the author has also made another significant change to the SATANS family of codes. In particular, the SATANS-II program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution, developed by Ryan [10] in 1972 to handle more complex and larger problems, was modified to make the computer memory requirement a variable quantity. This



quantity is specified by the user to fit the particular problem being run. It eliminates the large core requirement of SATANS-II for small problems and allows for much larger problems to be solved than could be solved by SATANS-II. The new program with the pole condition and memory modifications will hereafter be called SATANS-IIA. It is described in section II.



## II. DESCRIPTION OF SATANS-IIA

SATANS-II was developed by Ryan [10] from SATANS-I and incorporated the full trigonometric expansion of the applied load and solution vector, and introduced the handling of imperfections into the code. These modifications allow the analysis of shells under totally arbitrary loads, as well as imperfection studies on actual shells with measured imperfections [11]. Unfortunately, the original deck of cards for SATANS-II was destroyed. Professor Johann Arbocz of CALTECH had a listing of SATANS-II and punched a deck of cards with the changes to SATANS-I given in that listing. A copy of this deck was sent to Professor Ball. These cards have been added by the author to the original SATANS-I described by Ryan [10] and a complete version of SATANS-II has been reconstructed. SATANS-IIA is a modification by the author of the reconstructed SATANS-II program. A listing of SATANS-IIA can be found in Appendix A. The listing contains an example problem for the dynamic analysis of a clamped, truncated cone subjected to an impulsive loading which is uniform along the meridian and varies in a cosine distribution over one-half of the circumference. This problem is a sample problem suggested by the Lockheed Missiles and Space Corp. [12]. A condensed version of the output from the example problem is given in Appendix B. Input data preparation for SATANS-IIA can be found in Appendix C. The basic users manual, which includes preparation of input subroutines and the theory of the program, is contained in Reference 3, which can be obtained through COSMIC (M70-10098, LAR-10736), or ASIAC [13]. A users manual which includes preparation and handling of imperfection data within the SATANS programs can be found in



Ref. [10]. The above information, along with the following discussion, will inform the user on the capabilities and proper use of SATANS-IIA.

The modification of the SATANS-II program to make its core requirement variable was accomplished by putting in a single dimension statement at the beginning of the program, with subsequent dimensioning within the subroutines to only the first element of the vector or matrix. This is a convenient feature of the FORTRAN-IV language in which the program is written. The actual vector and matrix sizes are transmitted to the subroutines by an individual parameter list. Construction of the initial dimension statement and core request size is as follows:

The basic size of the program on the IBM-360/67 Digital Computer, without the initial dimension statement, is 272,000 bytes. This figure includes approximately 19,000 bytes of buffer space required for execution. Within the main dimension statement are fifteen variables. However, only three parameters are needed to specify the sizes of these fifteen variables.

Let a= The number of stations along the meridian of the shell times the number of harmonics considered.

Let b= a, plus two fictitious stations times the number of harmonics considered.

Let c= The number of harmonics considered.

The main dimension statement would then be constructed as,

```
DIMENSION P(4,4,a), DEE(4,4,a), DST(4,4,a), X(4,a),  
           PHIXB(a), PHITB(a), Z(4,b), ZO(4,b),  
           Z2(4,b), Z3(4,b), ZDOT(4,b), IS(99,c),  
           JS(99,c), ID(99,c), JD(99,c)
```



The 99's above limit the user to 99 harmonics in any one run and an unlimited number of meridional stations. The core requirement for the general case would be,

$$272,000 + 216a + 80b + 1584c = \text{bytes of core required.}$$

For a sample calculation of the core requirements consider the example of a spherical cap with 40 stations along the meridian, and an asymmetric analysis with two harmonics. Therefore,

$$a = 40(\text{stations}) \times 2(\text{harmonics}) = 80$$

$$b = 80 + 2 \times 2(\text{harmonics}) = 84$$

$$c = 2(\text{harmonics})$$

Thus, for the variables P, DEE, DST,

$$3 \times (4 \times 4 \times 80) = 3840 \text{ (words)} \times 4 = 15,360 \text{ bytes}$$

for the variable X,

$$4 \times (80) = 320 \text{ (words)} \times 4 = 1280 \text{ bytes}$$

for the variables PHIXB, PHITB,

$$2 \times (80) = 160 \text{ (words)} \times 4 = 640 \text{ bytes}$$

for the variables Z, Z0, Z2, Z3, ZDOT,

$$5 \times (4 \times 84) = 1680 \text{ (words)} \times 4 = 6720 \text{ bytes}$$

lastly, for the variables ID, JD, IS, JS,

$$4 \times (99 \times 2) = 792 \text{ (words)} \times 4 = 3168 \text{ bytes}$$

Therefore, the total size of the main dimension statement would be 27,168 bytes. This figure would be rounded up to the nearest even thousand bytes, i.e. 28,000 bytes. Finally, the core requirement for this example problem would be

$$272,000 + 28,000 = 300,000 \text{ bytes.}$$



### III. IMPROVED POLE ROUTINE

The SATANS code is based upon Sander's geometrically nonlinear equations under the conditions of small strains and moderately small rotations. The formulation is in four second order nonlinear partial differential equations in terms of  $U$ ,  $V$ ,  $W$ , and  $M_s$ , where  $U$ ,  $V$ , and  $W$  are the meridional, circumferential and normal displacements respectively, and  $M_s$  is the meridional bending moment. The nonlinear partial differential equations in the coordinates  $s$ ,  $\theta$ , and  $t$  are reduced to uncoupled sets of linear differential equations in  $s$  and  $t$  by expanding the variables in trigonometric series in the circumferential coordinate  $\theta$ , and treating the nonlinear terms as pseudo loads. The first and second derivatives in the meridional coordinate  $s$  are replaced by the conventional central finite difference approximations, i.e.

$$\{z\}'_i = 1/2\Delta (\{z\}_{i+1} - \{z\}_{i-1}) \quad (1)$$

and

$$\{z\}''_i = 1/\Delta^2 (\{z\}_{i+1} - 2\{z\}_i + \{z\}_{i-1}) \quad (2)$$

where  $\{z\}_i$  is the vector of  $U$ ,  $V$ ,  $W$ , and  $M_s$  at the  $i^{\text{th}}$  station,  $\Delta$  is the uniform dimension between stations, and primes denote partial derivatives with respect to  $s$ . Applying these approximations to the governing set of domain



equations leads to

$$[C]_i \{z\}_{i-1} + [B]_i \{z\}_i + [A]_i \{z\}_{i+1} = \{g\}_i \quad (3)$$

When the shell does not have a pole, fictitious stations one increment off of the shell are introduced at each end. Both the governing domain equations and the boundary conditions are applied at the two boundary points. Thus, all finite difference approximations to the derivatives, including those of the boundary conditions, are of order

$\Delta^2$ . However, prior to the development of SATANS-IIA, the treatment of the conditions to be applied at a pole at either end of a shell was handled by a simple Euler forward or backward difference approximation to the first derivative, with truncation error of order  $\Delta$ . For example, for a pole at  $s=0$ , where  $i=1$ , the first derivative at the pole was approximated with

$$\{z\}'_1 = 1/\Delta (\{z\}_2 - \{z\}_1). \quad (4)$$

At the time this procedure for handling the pole conditions was developed (1967) it was thought that this would not significantly alter the solution. However, it has since been discovered that such is not the case.

For the new pole routine, an expanded forward difference approximation of order  $\Delta^2$  is used at  $s=0$  which takes into account the two stations after the pole, instead of just one station after the pole as in the Euler scheme. This approximation is

$$\{z\}'_1 = 1/2\Delta (-3\{z\}_1 + 4\{z\}_2 - \{z\}_3). \quad (5)$$



The conditions to be imposed upon the dependent variables at a pole are derived in Reference 14. They are :

$$\text{For } N=0, \quad u_1 = v_1 = w'_1 = m'_s = 0.$$

Applying equation (5), these conditions can be put into the matrix form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

where the above 3 matrices are DL, DG, and DF within the SATANS programs.

$$\text{For } N=1, \quad u_1 \pm v_1 = u'_s = w = m_s = 0,$$

where the plus sign applies at an initial pole, and the minus sign at a final pole. The matrix form for these conditions is

$$\begin{bmatrix} -3 & 0 & 0 & 0 \\ 1 \pm 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

$$\text{For } N=2, \quad u = v = w = m'_s = 0$$

the matrix form is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

$$\text{For } N > 2, \quad u = v = w = m_s = 0$$



and DL= identity matrix, DG= DF= null matrices.

The solution procedure in SATANS is an elimination scheme and starts with

$$\{z\}_1 = - [P]_1 \{z\}_2 + \{x\}_1, \quad (6)$$

where the values in  $[P]_1$  based upon the Euler approximation are defined in Reference 14. The higher order approximation defines a new  $[P]_1$ . This new  $[P]_1$  is obtained by simultaneously solving the pole conditions

$$[DL] \{z\}_1 + [DG] \{z\}_2 + [DF] \{z\}_3 = \{0\}, \quad (7)$$

and the domain equation at station 2 next to the pole

$$[C]_2 \{z\}_1 + [B]_2 \{z\}_2 + [A]_2 \{z\}_3 = \{g\}_2, \quad (8)$$

to eliminate  $\{z\}_3$ . Thus,

$$\{z\}_3 = [A]_2^{-1} (\{g\}_2 - [C]_2 \{z\}_1 - [B]_2 \{z\}_2). \quad (9)$$

Substituting equation (9) into equation (7) gives

$$[DL] \{z\}_1 + [DG] \{z\}_2 + [DF] [A]_2^{-1} (\{g\}_2 - [C]_2 \{z\}_1 - [B]_2 \{z\}_2) = 0. \quad (10)$$

Combining like coefficients of the  $\{z\}$  vector leads to

$$([DL] - [DF] [A]_2^{-1} [C]_2) \{z\}_1 + ([DG] - [DF] [A]_2^{-1} [B]_2) \{z\}_2 = - [DF] [A]_2^{-1} \{g\}_2. \quad (11)$$

Finally, solving for  $\{z\}_1$  yields



$$\{z\}_1 = - [DL - DF \times A_2^{-1} \times C_2]^{-1} [DG - DF \times A_2^{-1} \times B_2] \{z\}_2 \\ + [DL - DF \times A_2^{-1} \times C_2]^{-1} [-DF \times A_2^{-1}] \{g\}_2. \quad (12)$$

Thus,  $[P]_1 = - [DL - DF \times A_2^{-1} \times C_2]^{-1} [DG - DF \times A_2^{-1} \times B_2]$  and  $\{x\}_1 = [DL - DF \times A_2^{-1} \times C_2]^{-1} [-DF \times A_2^{-1}] \{g\}_2$ . The new  $[P]_1$  matrix has been placed into the "PMATRIX" subroutine of SATANS-IIA and the new  $\{x\}_1$  vector has been placed in the "FORCE" subroutine.

A listing of the pole routine may be found in Appendix D. To incorporate this new routine into a SATANS-I or-II program, first proceed to the "PMATRIX" subroutine and remove the fifteen cards that are between, but not including, "IF(NN.GT.2) GO TO 90" and "11 CONTINUE". These cards are located after statement number "14" and just before statement number "11". Replace the cards removed by the ones listed in Appendix D which read from "C IN PMATRIX" to "90 M3=MN". Then proceed to the "FORCE" subroutine and remove statement number "10". Replace statement number "10" with the nine cards listed in Appendix D which read from "C IN FORCE" to "DO 11 I= 1,4". Also place "COMMON /IBL5/IBCINL, IBCFNL" into the common area of the "FORCE" subroutine.

This completes the implementation of the new pole routine into either SATANS-I or II.



#### IV. PROBLEM DESCRIPTION

The geometry of the shallow spherical shell used in this study is identical to that used in Reference 1. Briefly, the shallow shell can be specified by the non-dimensional parameter  $\lambda$ , where

$$\lambda = 2[3(1 - \nu^2)]^{1/4} (H/h)^{1/2}. \quad (1)$$

H is the rise of the shell, h is the thickness, and  $\nu$  is Poisson's ratio. The mass density of the shell is  $m$ . All shells analyzed had the following dimensions;

Radii of Curvature	$R = R_s = 250$ inches
Thickness	$h = 0.25$ inches
Modulus of Elasticity	$E = 30,000,000$ psi
Poisson's Ratio	$\nu = 0.3$

All buckling pressures obtained will be listed as a percent of the classical buckling pressure of a complete sphere,  $q_0$ , where

$$q_0 = [2E(h/R_s)^2] / [3(1 - \nu^2)]^{1/2} \quad (2)$$

Forty stations were used over the meridian. The nondimensional time increment  $\delta t$ , where

$$t = T / (R_s^2 m / E)^{1/2}, \quad (3)$$



was taken as 0.05 for 3000 time steps, which is a total nondimensional time of 150. In addition, the axisymmetric analysis was repeated with a larger time step of  $\delta t = 0.2$  for a total time of 600. In this study  $m$  was selected such that  $t$  is equal to  $T$ . The necessity for the long response time is explained in Reference 6.

In the axisymmetric analysis only the  $N = 0$  harmonic is considered. However, in the asymmetric analysis a second harmonic is excited by applying an incremental load in that harmonic. In addition, analyses of the shells  $\lambda = 6, 7.5,$  and  $11$  were made using five harmonics. The step pressure load for the axisymmetric harmonic is

$$\{q^{(0)}\} = P q_0 \{1\}, \quad (4)$$

and the step pressure load for the asymmetric second harmonic is

$$\{q^{(n)}\} = P q_0 \xi^{(n)} \{1\}, \quad (5)$$

where  $n > 0$ , and  $\xi^{(n)}$  is taken as 0.0001. The value taken for the second harmonic in the asymmetric analysis was the same as the critical harmonic for the static buckling analysis presented by Stilwell and Ball [2]. When there was an uncertainty as to which was the critical static harmonic the two harmonics in question were both tested. Run times using SATANS-IIA with a two-harmonic analysis for 3000 time steps and 40 stations on the meridian took an average of 28 minutes on the IBM 360/67.

The parameter used to determine the minimum load at which dynamic buckling occurs is the peak value of  $\bar{V}$ , called



$\bar{V}_{MAX}$ , where  $\bar{V}$  is defined as

$$\bar{V} = \int_0^{r_0} r w^{(0)} dr / \int_0^{r_0} r \xi dr \quad (6)$$

$r$  is the normal distance from the axis to the shell,  $r_0$  is the maximum value of  $r$ ,  $w^{(0)}$  is the normal displacement of the axisymmetric response and  $\xi$  is the vertical distance from the base plane to the undeformed shell. The  $\bar{V}$  is a measure of the volume of the shell deformation. The Fortran statements computing  $\bar{V}$  and  $\bar{V}_{MAX}$  are given in Appendix E.

When working a problem that requires these calculations the nineteen cards are inserted directly into the "DYNAMIC" subroutine right after the "IF" statement that calls the "OUTPUT" subroutine.

For convenience, the response in each asymmetric harmonic is also measured using equation (6), with  $w^{(0)}$  replaced with  $w^{(n)}$ . The parameter  $\bar{V}$  for the asymmetric harmonics does not represent a volume of deformation as it does for the axisymmetric harmonic. It can, however, be used to indicate the relative excitation of the asymmetric harmonics.

The buckling criterion for both the axisymmetric and the asymmetric dynamic buckling analysis defines the critical load as that load  $P$  where a very small increase in  $P$  causes a very large increase in  $\bar{V}_{MAX}$ . This is the same criterion



as that used in Ref. [1].



## V. RESULTS AND DISCUSSION

### A. STATIC AXISYMMETRIC BUCKLING ANALYSIS

Table I presents the new results from the static axisymmetric buckling analyses for  $\lambda = 4$  through 13 using the new pole routine. The two upper curves in Figure 1 present a comparison of the new results obtained by SATANS-IIA with those obtained by Stilwell and Ball [2] using the SATANS-I program. As can be seen in this figure, fairly significant changes in the buckling load occurred in the neighborhood of  $\lambda = 4, 5$ , and 9; and somewhat smaller differences occurred in the region  $\lambda = 10$  through 13. The upper data points in Figure 2 present the comparison of the new results from SATANS-IIA with those obtained by Huang [4]. This comparison shows a very good agreement between the two sets of results, except for the largest values of  $\lambda$ . The new results have eliminated the differences that existed between the SATANS-I results and Huang's results.

### B. DYNAMIC AXISYMMETRIC BUCKLING ANALYSIS

Figure 3 presents the new results for the peak value of  $\bar{V}_{MAX}$  versus P for the various values of  $\lambda$  tested. Table II presents all of the new results for the dynamic axisymmetric buckling load. These loads are selected from figures constructed just like Figure 3. In every case,



except for  $\lambda = 4$ , a value of  $P$  slightly above the  $P_{CRIT}$  value caused a  $\bar{v}_{MAX}$  indicative of buckling, as well as a nonconvergence of the iterative solution procedure.

The lower two curves of Figure 1 present a comparison versus  $\lambda$  of the new axisymmetric dynamic buckling results with the previous buckling results obtained by Ball and Burt [1]. In every case the new critical pressure is lower than the critical pressure obtained using the Euler approximation at the pole.

The lower data points of Figure 2 present a comparison of the new results with those obtained by Huang [5], by Stephens and Fulton [6], and by Stricklin [8]. Just as in the case of the static axisymmetric buckling analysis, the new results compare much more favorably with the other results than did the results of Reference 1. It's interesting to note that the new results now tend to be slightly lower than the other results, whereas the results of Reference 1 were higher for almost all values of  $\lambda$ .

### C. DYNAMIC ASYMMETRIC BUCKLING ANALYSIS

Table III presents the new results for the critical pressures obtained from the dynamic asymmetric analysis. The second harmonics, or critical static harmonics, used in the analyses are also presented in Table III. A comparison of the critical pressures from the asymmetric analyses, Table III, with the critical pressures from the axisymmetric analyses, Table II, reveals that only the shell  $\lambda = 6$  buckled at a load below the axisymmetric buckling load. For the shell  $\lambda = 7$  the critical buckling load was slightly



larger when asymmetric motion was considered. In all other cases the buckling was not influenced by the presence of the second harmonic. These new buckling results and those by Ball and Burt [2] are plotted in Figure 4. The new results can be seen to be significantly different from the SATANS-I results, where the asymmetric buckling loads were lower than the axisymmetric loads for five out of the ten values of tested.

Except for  $\lambda = 6$  and 7, the relationship between  $\bar{V}_{MAX}$  and P for the N= 0 harmonic, in the two-harmonic analyses, was found to be essentially identical to the relationship found in the axisymmetric buckling analysis shown in Figure 3. Table IV A presents the  $\bar{V}_{MAX}$  versus P data for both the N= 0 harmonic and the second harmonic, for all values of  $\lambda$  tested, except for  $\lambda = 6$ . Note that, except for  $\lambda = 7$  and 11,  $\bar{V}_{MAX}$  for the asymmetric harmonic is generally very small, even when the  $\bar{V}_{MAX}$  for the N= 0 harmonic indicates that the shell has buckled. Thus, except for the shells  $\lambda = 6$  and 7, the presence of the asymmetric motion does not influence the axisymmetric motion, and except for the shells  $\lambda = 6, 7$  and 11 the asymmetric motion is very small prior to buckling in the axisymmetric harmonic.

A more detailed analysis of the shell  $\lambda = 6$  has been conducted since it was the only shell that revealed any significant axisymmetric sensitivity to asymmetric motion. This shell was studied using two two-harmonic analyses (N= 0, 1 and N= 0, 2) and a five-harmonic analysis (N= 0, 1, 2, 3, and 4). Figure 5 and Tables IV B and IV C contain values of  $\bar{V}_{MAX}$  versus P for both of the asymmetric harmonics, N= 1



and  $N = 2$ , in the two two-harmonic analyses, as well as the values of  $\bar{V}_{MAX}$  for the axisymmetric harmonic,  $N = 0$ . Figure 6 and Table IV D present the values of  $\bar{V}_{MAX}$  versus  $P$  for the  $N = 0, 1, 2, 3$ , and 4 harmonics from the five-harmonic study. A comparison of the critical buckling load predicted from the results of the two two-harmonic analyses in Figure 5 with the critical load from the five-harmonic analysis obtained from Figure 6 shows that the presence of the additional harmonics results in the shell buckling at a slightly lower load (0.50), with significant motion in the  $N = 1$  harmonic instead of the  $N = 2$  harmonic (see the nonconverged solution at  $P = 0.51$ ), which is the critical harmonic for static asymmetric buckling. Studies using five harmonics have also been conducted for  $\lambda = 7.5$  and  $\lambda = 11$ . As can be seen in Table IV D the critical harmonic for  $\lambda = 7.5$  remained  $N = 3$ ; however, significant motion occurred in that harmonic at  $P = .41$  and  $.44$ . In the case of  $\lambda = 11$ , relatively large asymmetric motion occurred in the asymmetric mode of  $N = 5$  vice 6 at a value of  $P = .46$ .

The comparison of the new results for the critical pressure for dynamic asymmetric buckling with those obtained analytically by Stricklin [8], by Akkas [9], and experimentally by Lock et al [7] is illustrated in Figure 7. The comparison reveals an agreement with Stricklin in every case, in general a higher value of  $P_{CRIT}$  than those obtained by Akkas, and most importantly a very good agreement with Lock's experimental results.

When making the comparison between the new results and those obtained by Akkas, it is necessary to look at the differences in the problem solution parameters used in the two studies. For example, buckling results obtained from SATANS-IIA using the same time increment as used by Akkas,



$\delta t = .2$  for 3000 time steps, were significantly higher than those using the time step of  $\delta t = .05$  for many values of  $\lambda$ . Furthermore, the new results had, in some cases, instances of buckling occurring as far out in time as 130. Akkas, to shorten computer run times, observed the cap only for a time of less than 5. Furthermore, only the harmonics  $N = 1$  or 2 or 3 were studied by Akkas for shells  $\lambda = 5$  through 12. If the critical harmonic is not studied, the predicted load will be too high. Thus, it appears that Akkas' lower bound loads may not be true lower bounds.

Two additional features of the shell response should be noted. First, shells  $\lambda = 6, 7.5,$  and 11 exhibited a non-buckled response in the axisymmetric harmonic to a load larger than the defined critical buckling load. This can be seen in Tables IV A and IV C. Second, and most importantly, the buckling load proposed by Ball and Burt [1], and used here, defines buckling to occur when the  $\bar{V}_{MAX}$  in the axisymmetric harmonic undergoes a large change due to a small change in  $P$ . Another criterion for dynamic buckling in the asymmetric analysis discussed in Reference 1 is to define the buckling load as that threshold load that initiates significant growth in the asymmetric harmonic. Re-examination of the  $\bar{V}_{MAX}$  versus  $P$  data in Table IV A through D reveals that shells  $\lambda = 6, 7,$  and 11 exhibited relatively large asymmetric motion at loads smaller than the defined buckling load when compared with other  $\bar{V}_{MAX}$  values for those shells, even though the numbers themselves were small when compared with the axisymmetric harmonic. Shells  $\lambda = 7.5$  and 12 appear to be borderline cases. If the alternate criterion for buckling is used, the critical buckling loads for shells  $\lambda = 6, 7,$  and 11 become 0.47, 0.45, and 0.45, respectively. The shells  $\lambda = 7.5$  and 12 could have buckling



loads as low as 0.40 and 0.44, respectively. These values are more conservative than the definition based upon axisymmetric response. These five shells are the same five shells that exhibited an asymmetric buckling load lower than the axisymmetric buckling load in Reference 1.



## VI. SUMMARY AND CONCLUSIONS

A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program, called SATANS-IIA, was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and dynamic nearly axisymmetric step-pressure loads of infinite duration. The cap sizes ranged from  $\lambda = 4$  to 13 including  $\lambda = 7.5$ . A comparison was made between the new buckling results with the improved pole handling routine and the results that did not have the new pole routine. The comparison revealed a significant change in buckling pressures, due solely to the change from an order  $\Delta$  finite difference approximation of the first derivatives at the pole to an approximation of order  $\Delta^2$ . These new critical pressures are in very good agreement with the results from other studies of the same spherical shells. This good agreement with other results, which came about as a result of the modification of the pole handling routine, is a strong indication that the manner in which the pole condition is handled is vital to the accuracy of the solutions obtained.

In the asymmetric analysis, two harmonics were included for most of the shells; the axisymmetric harmonic and one asymmetric harmonic. Five-harmonic analyses were conducted for three of the shells. Two buckling criteria for the



asymmetric analysis were considered. One defined buckling as that threshold load that caused a large increase in a deformation parameter,  $\bar{V}_{MAX}$ , in the axisymmetric harmonic.

The other, more conservative than the first, defined buckling as that threshold load that caused a large increase in the  $\bar{V}_{MAX}$  value for the asymmetric harmonic. Both values have been presented.

The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells. The effect of realistic imperfections remains to be determined.



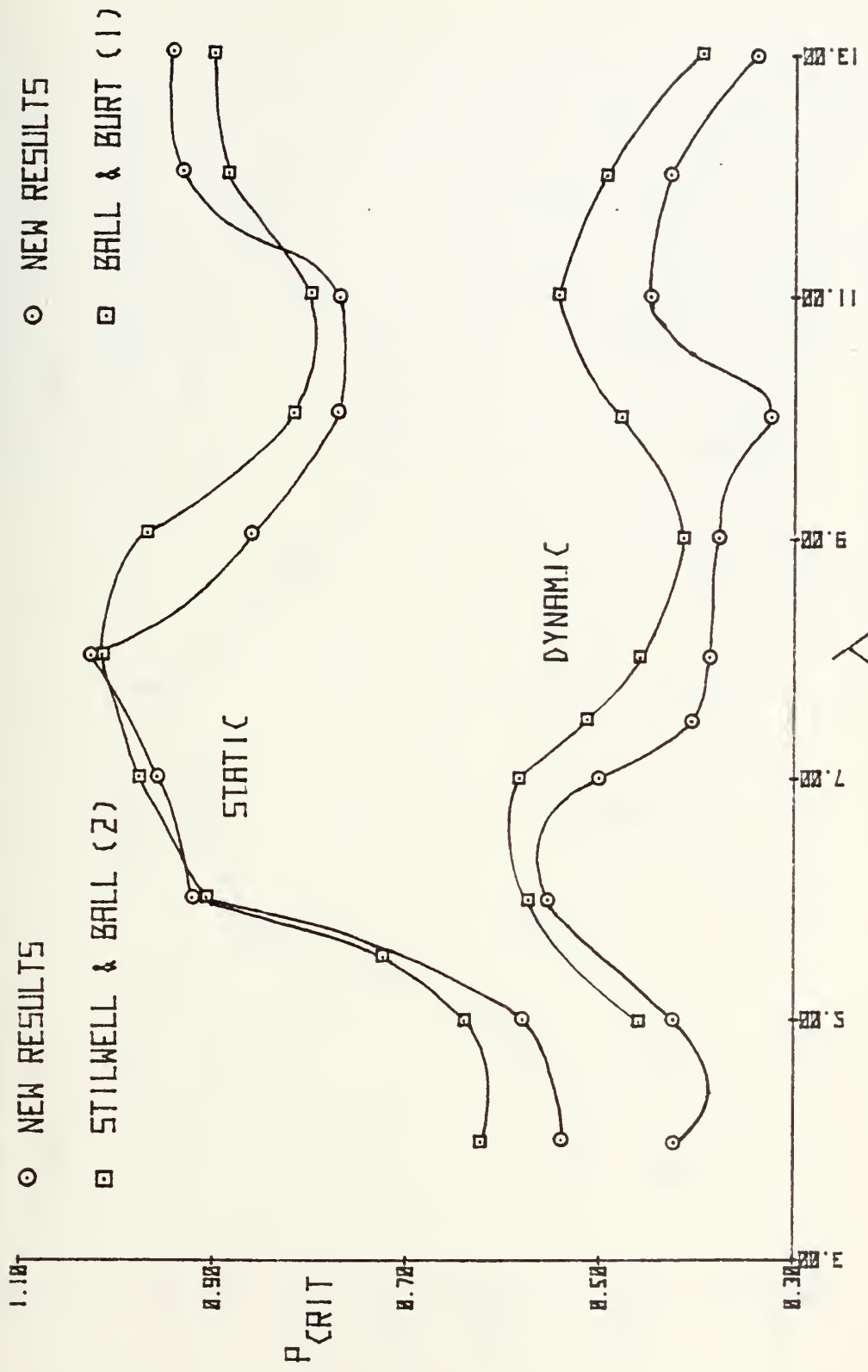


Figure 1 - CRITICAL STEP-PRESSURE LOAD VERSUS  $\lambda$   
 AXISYMMETRIC (SATANS-I VERSUS SATANS-IIA)



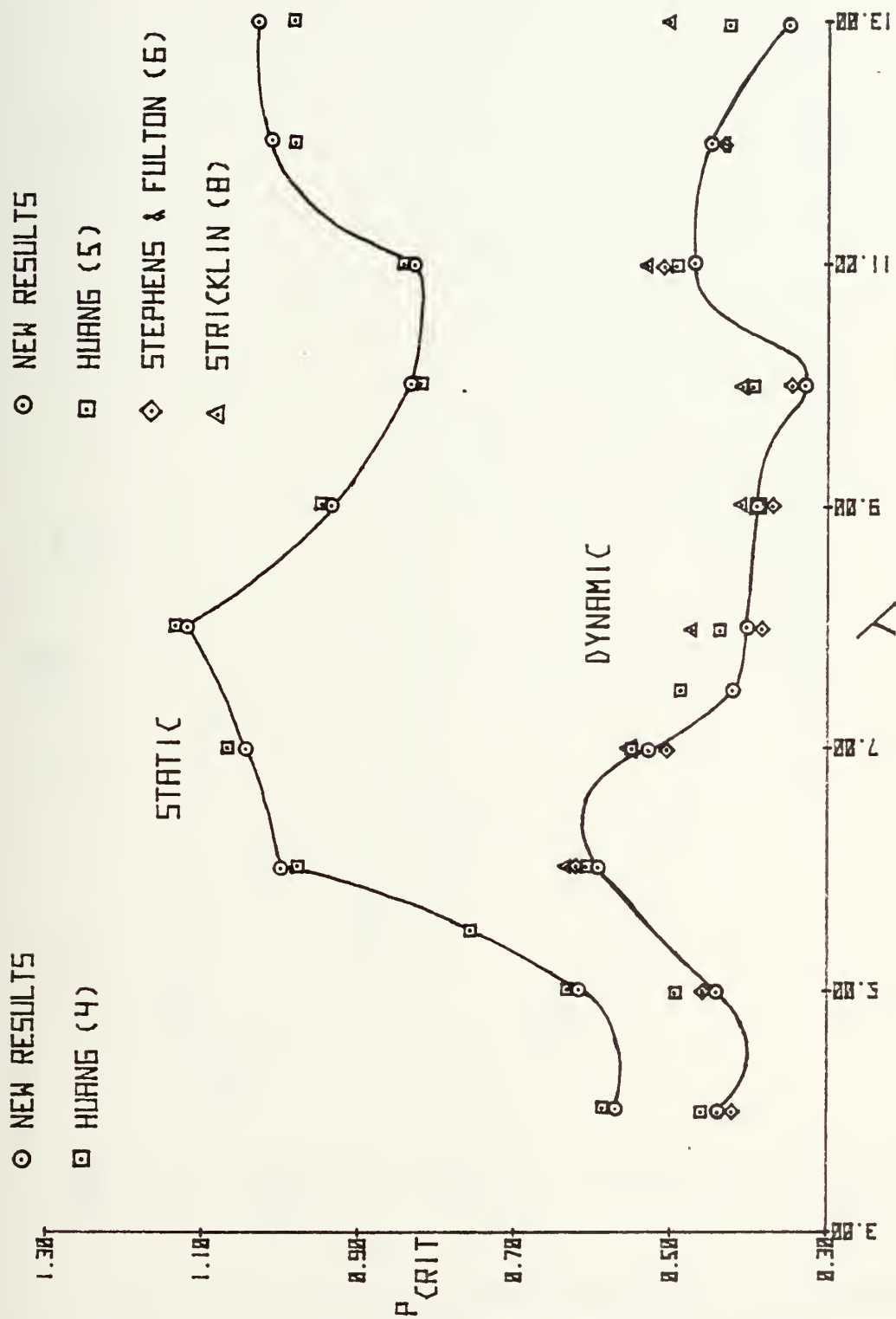


Figure 2 - CRITICAL STEP-PRESSURE LOAD VERSUS  $\lambda$   
AXISYMMETRIC (SATANS-IIA VERSUS ALL OTHERS)



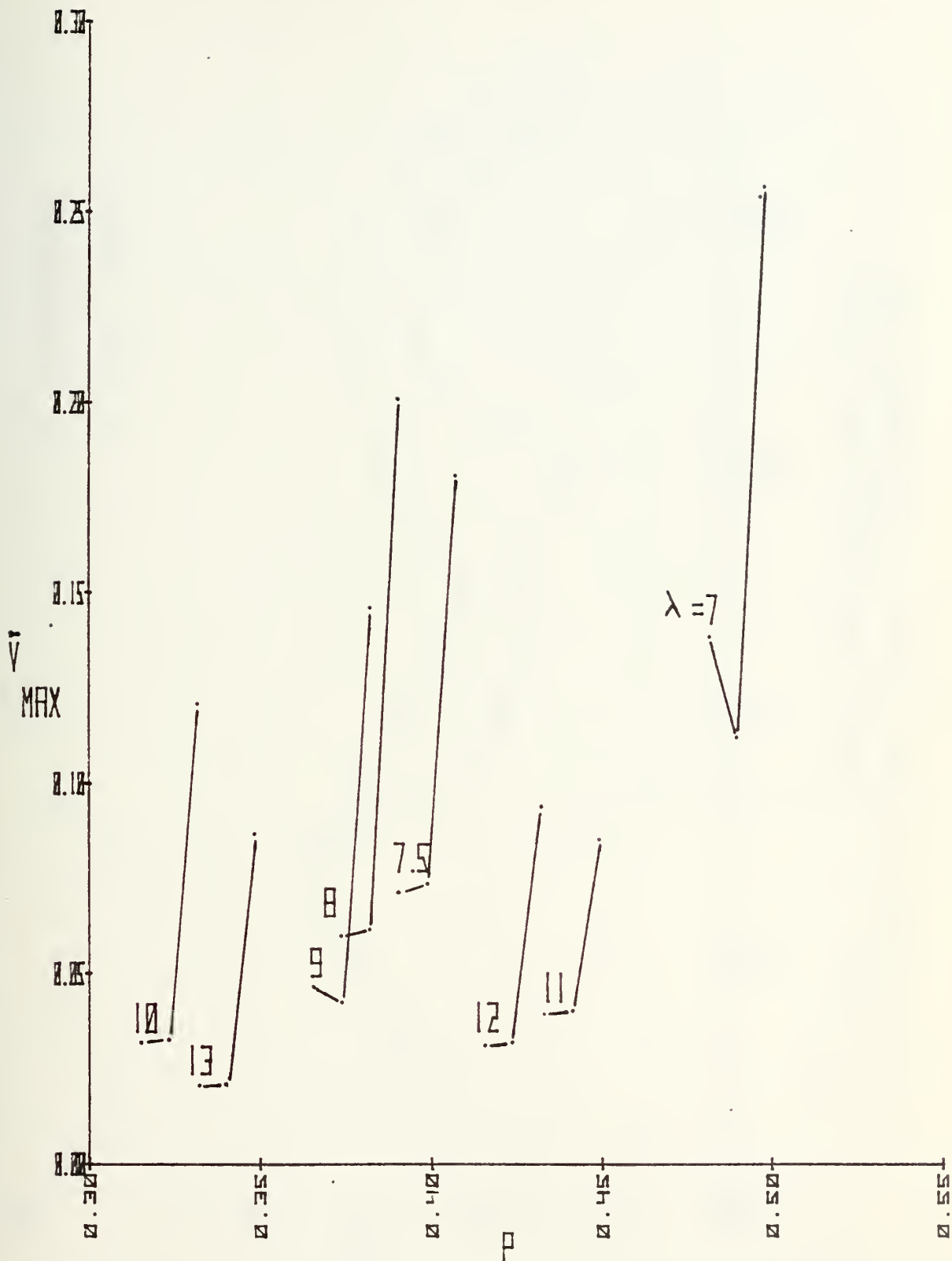


Figure 3 - PEAK DEFLECTION VERSUS  $P$ , AXISYMMETRIC AND ASYMMETRIC CASES FOR VARIOUS VALUES OF  $\lambda$  (SATANS-IIA)



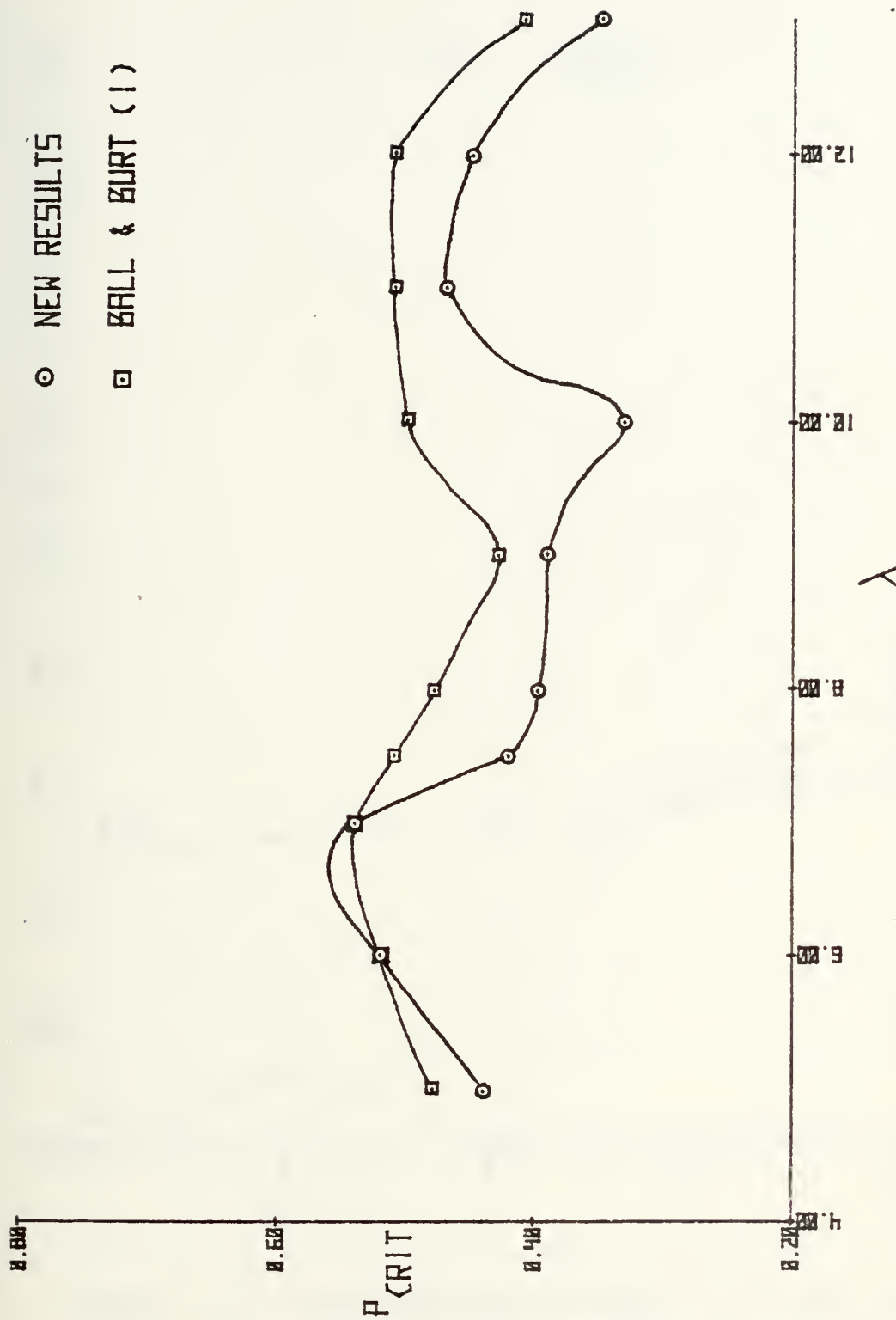


Figure 4 - CRITICAL STEP-PRESSURE LOAD VERSUS  $\lambda$   
ASYMMETRIC ANALYSES (SATANS-I VERSUS SATANS-IIA)





Figure 5 - PEAK DEFLECTION VERSUS P FOR THE ASYMMETRIC ANALYSES OF  $\lambda = 6$  ( $N=0,1$  AND  $N=0,2$ )



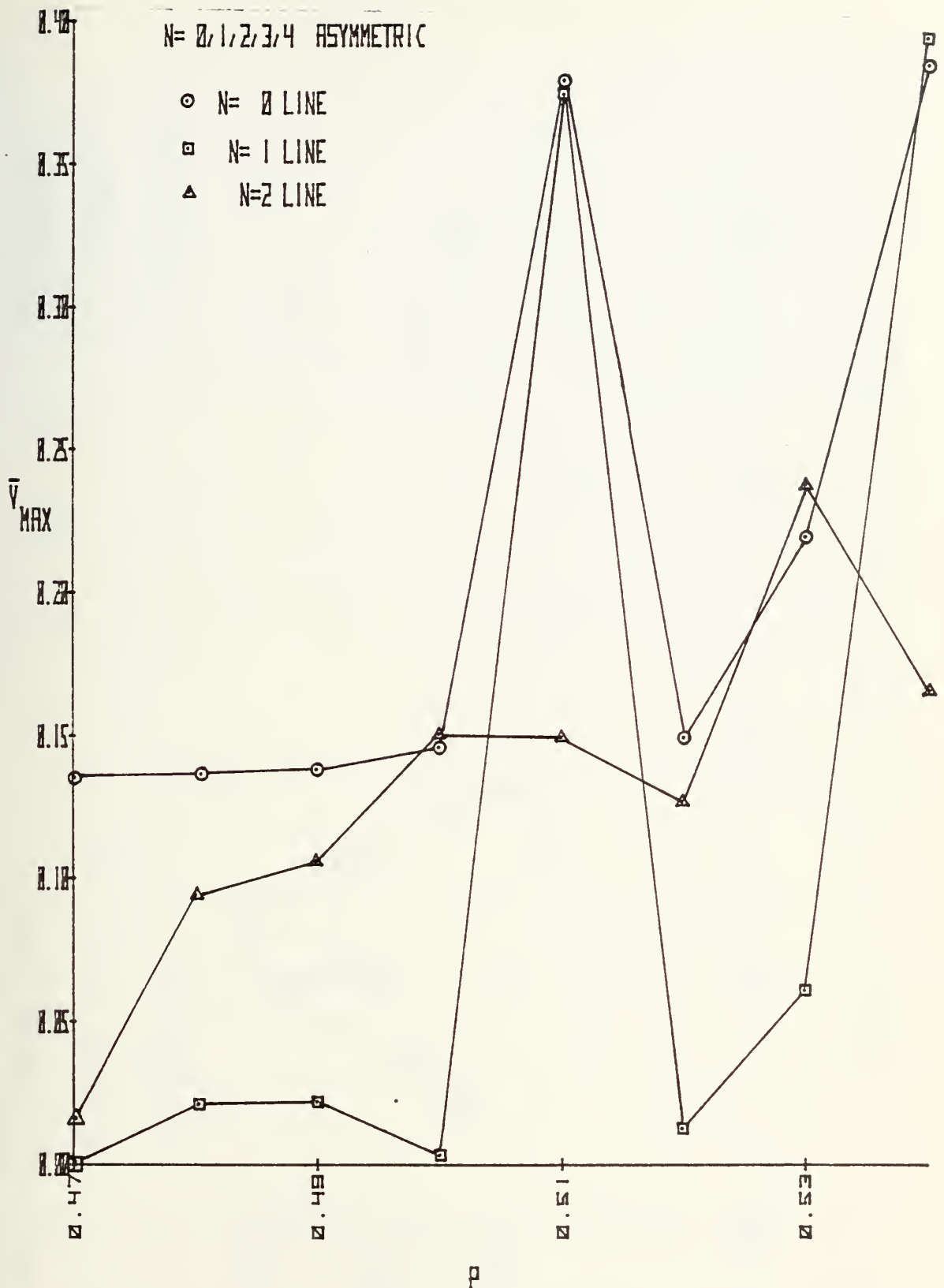


Figure 6 - PEAK DEFLECTION VERSUS P FOR THE ASYMMETRIC ANALYSES OF  $\lambda = 6$  (N=0,1,2,3,AND4, ONLY N=0,1,AND2 PLOTTED)



NEW RESULTS

○ ASYMMETRIC

— AXISYMMETRIC

I STRICKLIN (8)

□ LOCK, ET AL. (7)

▽ AKKAS (9)

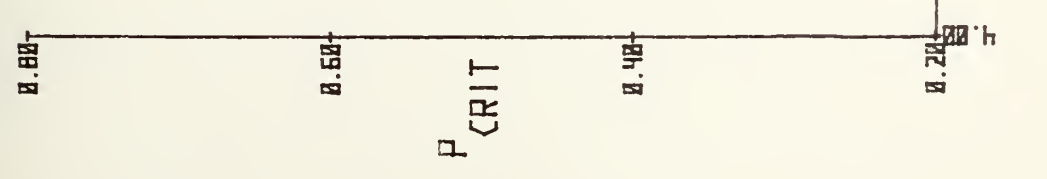


Figure 7 - CRITICAL STEP-PRESSURE LOAD VERSUS  $\lambda$   
ASYMMETRIC ANALYSES (SATANS-IIA VERSUS ALL OTHERS)



## A. TABLES

1. TABLE I Critical pressure loads from the static axisymmetric analyses.

$\lambda$	4	5	6	7	8	9	10	11	12	13
$P_{CRIT}$	.568	.616	1.0	1.048	1.12	.936	.832	.832	1.016	1.032

2. TABLE II Critical step-pressure loads from the axisymmetric dynamic analyses.

$\lambda$	4	5	6	7	7.5	8	9	10	11	12	13
$P_{CRIT}$	.45	.44	.59	.53	.42	.40	.39	.33	.47	.45	.35

3. TABLE III Critical step-pressure loads from the dynamic asymmetric analyses and critical asymmetric harmonics.

$\lambda$	5	6	7	7.5	8	9	10	11	12	13
$P_{CRIT}$	.44	.52	.54	.42	.40	.39	.33	.47	.45	.35
$N_{CRIT}$	1	2	3	3	4	5	6	7	8	9



4. TABLE IV Dynamic asymmetric analyses for  $\bar{v}_{MAX}$  versus

P.

1. TABLE IV A. Two-harmonic analyses for all values of  $\lambda$  except  $\lambda = 6$ .

$\lambda = 5$  N= 0 and 2

N= 0 and 1

P	.43	.44	.45	P	.44	.45
N= 0	.1659	.1676	.6606	N= 0	.1675	.6606
N= 2	.0004787	.0000566	.0687	N= 1	.0003145	.0687
				P	.46	
				N= 0	.7653	
				N= 1	.001092	

$\lambda = 7$ , N= 0 and 3

P	.45	.46	.47	.48	.49	.50	.52
N= 0	.09452	.09571	.09812	.1005	.1029	.1052	.1099
N= 3	.000889	.007456	.05052	.04323	.0279	.0335	.07488
P	.53	.54	.55				
N= 0	.1122	.1146	.2709				
N= 3	.05997	.06252	.03809				

$\lambda = 7.5$ , N= 0 and 3

P	.40	.41	.42	.43	.44	.45
N= 0	.0703	.07228	.07429	.2636	.07837	.2076
N= 3	.0001094	.001296	.0004304	.002754	.001188	.000338
P	.46					
N= 0	.200					
N= 3	.0003276					



$\lambda = 8$ ,  $N = 0$  and 4

P	.38	.39	.40	.41	.42	.43
N= 0	.05893	.0607	.0624	.1964	.1713	.1957
N= 4	.0000566	.0000703	.0000364	.0000333	.0000274	.0000299
P	.44					
N= 0	.2297					
N= 4	.0000326					

$\lambda = 9$   $N = 0$  and 4

$N = 0$  and 5

P	.38	.39	.40	P	.40
N= 0	.04738	.04875	.1576	N= 0	.05012
N= 4	.00003597	.00004635	.00004497	N= 5	.00008385

$\lambda = 10$ ,  $N = 0$  and 5

P	.32	.33	.34	.36	.38	.40
N= 0	.03239	.03347	.1086	.1217	.1288	.1235
N= 5	.0000281	.0000472	.00004125	.00002103	.0000449	.000114

$\lambda = 11$ ,  $N = 0$  and 6

P	.45	.46	.46	.48	.49	.50
N= 0	.03910	.04004	.04099	.09814	.04241	.08824
N= 6	.004595	.01332	.02232	.02864	.03955	.02813

$\lambda = 12$ ,  $N = 0$  and 7

P	.44	.45	.46
N= 0	.03236	.03316	.08633
N= 7	.00004214	.0004561	.00005158

$\lambda = 13$ ,  $N = 0$  and 8

P	.34	.35	.36	.38	.40
N= 0	.02119	.02185	.06637	.07844	.07381
N= 8	.00001148	.00001134	.000006607	.000008245	.000119



2. TABLE IV B. Two-harmonic analyses with  $N=0$  and  $1$ ,  
 $\lambda = 6$  only.

P	.48	.49	.50	.51	.52	.53
N= 0	.1350	.1385	.1421	.1460	.1499	.4453
N= 1	.0002797	.000195	.000245	.000926	.04081	.3668

3. TABLE IV C. Two-harmonic analyses with  $N=0$  and  $2$ ,  
 $\lambda = 6$  only.

P	.44	.45	.46	.47	.48	.49	.50
N= 0	.1218	.1250	.1276	.1320	.1350	.1385	.1479
N= 2	.000239	.0101	.0293	.01976	.08768	.1223	.1419
P	.51	.52	.53	.54	.55	.56	
N= 0	.1526	.1499	.5137	.5060	.2040	.5305	
N= 2	.1654	.1816	.3878	.3996	.2156	.3617	

4. TABLE IV D Five-harmonic analyses for selected  
shells.

$\lambda = 6$   $N=0,1,2,3$ , and  $4$

P	.47	.48	.49	.50	.51	.52	.53
N= 0	.1313	.1347	.1382	.1460	.3797	.1498	.2200
N= 1	.00021	.02108	.02215	.003676	.3743	.01276	.0616
N= 2	.0187	.0953	.1069	.1507	.1502	.1279	.2385
N= 3	.000181	.006237	.01437	.00163	.0405	.0123	.03978
N= 4	.0031	.04757	.05428	.04402	.05896	.0495	.064
P	.54						
N= 0	.3854						
N= 1	.3953						
N= 2	.1671						
N= 3	.05298						
N= 4	.0613						



$\lambda = 7.5$   $N = 0, 1, 2, 3, 4$  and 4

P	.40	.41	.42	.43	.44	.45
N= 0	.0703	.07228	.07429	.2592	.07837	.2544
N= 1	.00004855	.00004198	.00006093	.0002737	.0001167	.005952
N= 2	.0001164	.00007456	.0004184	.0000982	.000788	.0003188
N= 3	.0001277	.001187	.0004597	.0002853	.00107	.0003188
N= 4	.0008224	.0001898	.0002448	.0000526	.000280	.000134

$\lambda = 11$   $N = 0, 4, 5, 6,$  and 7

P	.45	.46	.47	.48	.49	.50
N= 0	.03910	.04004	.0499	.04195	.04291	.1040
N= 4	.0005759	.001263	.0009774	.001388	.002657	.001565
N= 5	.009568	.0140	.0124	.02239	.02759	.01548
N= 6	.002560	.007828	.02330	.02767	.02602	.02644
N= 7	.0001743	.0001486	.0002021	.01202	.02048	.02064



APPENDIX A

LISTING OF SATANS-IIA







```

C ***** THIS SUBROUTINE COMPUTES THE NCNDIMENSIONAL IN-PLANE AND
C BENDING STIFFNESSES OF THE SHELL *****
C ***** REAL NU, LAM2, JAY, MT, LSD18, LSC1N *****
C ***** CCMMCN /BL15/ NU, U1(99), V1(99), W1(99), U2(99), U3(99),
1 CCMMCN /BL17/ V3(99), W3(99)
C ***** CCMMCN /BL32/ TKN, ELAST, CHAR, SIGC
C ***** B=1.C89082
C ***** L=.C5C15683
C ***** DE=C.
C ***** LC=0.
C ***** RETURN
C ***** END

```

```

C ***** SLRROUTINE PLOAD(K,Z) *****
C ***** THIS SUBROUTINE ESTABLISHES THE NON-DIMENSIONAL FCURIER
C ***** COEFFICIENTS OF THE LOADS APPLIED TO THE SHELL *****
C ***** REAL MASS *****
C ***** DIMENSION Z(4,1) *****
C ***** CCMMCN /IBL1/ MNMAX *****
C ***** CCMMCN /IBL2/ NN(99), MNINIT *****
C ***** CCMMCN /IBL4/ KMAX, KL *****
C ***** CCMMCN /IBL8/ LSTEP, ITR *****
C ***** CCMMCN /BL3/ PR(99), PX(99), PT(99) *****
C ***** CCMMCN /BL6/ SOE, OSE, ALOAD *****
C ***** CCMMCN /BL8/ R(500), GAM(500), QMT(500) *****
C ***** CCMMCN /BL32/ TKN, ELAST, CHAR, SIGC *****
C ***** CCMMCN /BL102/ DELCAD *****
C ***** CCMMCN /BL103/ MASS(500) *****
C ***** CCMMCN /BL17/ DEL/BL100/TEEO, $DYNMC *****
C ***** CCMMCN /BLTHTA/ TETAM, COEFF *****
C ***** RETRN *****
C ***** END *****

```

```

C ***** SLRROUTINE INITL (Z, ZC, Z2, Z3, ZCCT) *****
C ***** THIS SUBROUTINE DESCRIBES THE INITIAL CONICITIONS FOR DYNAMIC CASES *****
C ***** IMPLICIT LOGICAL*1 ($) *****
C ***** DIMENSION Z(4,1), ZC(4,1), Z2(4,1), Z3(4,1), ZCCT(4,1) *****
C ***** CCMMCN /IBL1/ MNMAX *****
C ***** CCMMCN /IBL2/ NN(99), MNINIT *****
C ***** CCMMCN /IBL4/ KMAX, KL *****
C ***** CCMMCN /IBL5/ MAXM *****
C ***** CCMMCN /IBL12/ KMAX1, KMAX2, NCONV *****
C ***** CCMMCN /BL6/ SOE, CSE, ALOAD *****

```



ROUTINE SATANS (P,CEE,DST,X,Z,ZC,Z2,Z3,ZCCT,IS,JS,ID,JD,PH,IXB,











```

IF (IPS) WRITE (6,225)
IF (IPT) WRITE (6,226)
IF (ITT) WRITE (6,227)
IF (IMT) WRITE (6,228)
IF (ICT) WRITE (6,229)
IF (IDMT) WRITE (6,230)
IF (IC+CK1=IABS(INS)+IABS(INTH)+IABS(IQS)+IABS(IMS)
1 +IABS(IMTH)+IABS(IMSTH)+IABS(IU)+IABS(IV)+IABS(IW)
2 +IABS(IPHIS))+IABS(IPHIT)+IABS(IPHI)
IF ((ICHCK1.GT.0).AND.$MCDAL) WRITE (6,231)
IF ((ICHCK1.EQ.0).AND.$MCDAL) WRITE (6,232)
IF (INS) WRITE (6,233)
IF (INT) WRITE (6,234)
IF (INSTH) WRITE (6,235)
IF (IQS) WRITE (6,236)
IF (IMS) WRITE (6,237)
IF (IMSTH) WRITE (6,238)
IF (IMTH) WRITE (6,239)
IF (IU) WRITE (6,240)
IF (IV) WRITE (6,241)
IF (IW) WRITE (6,242)
IF (IPHIS) WRITE (6,243)
IF (IPHIT) WRITE (6,244)
IF (IPHI) WRITE (6,245)
5 GC TO 6
5 WRITE (6,246) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,EPS
5 WRITE (6,247) CHAR,TKN,ELAST,SIGC,TEEO,NL
6 IF (NTHMAX.EQ.0) GC TO 7
7 WRITE (6,248)
7 WRITE (6,249) (TH(NTH),NTH=1,NTHMAX)
7 IF (NDIMEN.EQ.1) WRITE (6,250)
7 IF (NDIMEN.EQ.0) WRITE (6,251)
C** CALL APPROPRIATE CCROLLING SUBPRCGRAM *****
C** IF ($DYNMC) CALL DYNMC (P,CEE,CST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS,JD, *****
C** IPHIXB,PHITB) ***** ***** ***** ***** *****
C** IF (.NOT.$DYNMC) CALL STATIC (P,CEE,DST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS, *****
C** ILL,JD,PHIXB,PHITB) ***** ***** ***** ***** *****
C** REAL STATEMENT FORMATS FOLLOW ***** ***** ***** ***** *****
C** ***** ***** ***** ***** ***** ***** ***** ***** *****
1CC FCRMAT (18A4)
101 FCRMAT (15,L5,14I5)
102 FCRMAT (6E12.4)
103 FCRMAT (2E12.4)
104 FCRMAT (6E12.4)
SAT02290
SAT02300
SAT02310
SAT02320
SAT02330
SAT02340
SAT02350
SAT02360
SAT02370
SAT02380
SAT02390
SAT02400
SAT02410
SAT02420
SAT02430
SAT02440
SAT02450
SAT02460
SAT02470
SAT02480
SAT02490
SAT02500
SAT02510
SAT02520
SAT02530
SAT02540
SAT02550
SAT02560
SAT02570
SAT02580
SAT02590
SAT02600
SAT02610
SAT02620
SAT02630
SAT02640
SAT02650
SAT02660
SAT02670
SAT02680
SAT02690
SAT02700
SAT02710
SAT02720
SAT02730
SAT02740
SAT02750

```



```

1C5 FCRMAT (4E16.8)
1C6 FCRMAT (2L2,29I2)
1C7 FCRMAT (I2)
C** *****
C** WRITE STATEMENT FCRMATS FOLLOW
C** *****
2C0 FCRMAT (1,48X,---PRCBLEUM NUMBER,I5,---,///)
2C1 FCRMAT (1,38X,18A4,///)
2C2 FCRMAT (1,49X,---INPUT DATA RECCD---,///)
2C3 FCRMAT (1,110,THE SHELL HAS AN INITIAL PCLE,/)
2C4 FCRMAT (1,110,AT THE INITIAL EDGE,//2X,-----CMEGA BARSAT02880
1AR-----,15X,-----LAMDA BAR-----,1SAT02890
22X,---EL-----,/)
2C5 FCRMAT (1,1,4E10.3,) NS (1,4E10.3,) U (1,E10SAT02910
1.3)
2C6 FCRMAT (1,1,4E10.3,) NST + (1,4E10.3,) V = (1,E10SAT02930
1.3)
2C7 FCRMAT (1,1,4E10.3,) CS (1,4E10.3,) W (1,E10SAT02940
1.3)
2C8 FCRMAT (1,1,4E10.3,) PHIS (1,4E10.3,) MS (1,E10SAT02970
1.3)
205 FCRMAT (1,110,THE SHELL HAS A FINAL PCLE,/)
21C FCRMAT (1,110,AT THE FINAL ECCE,//2X,-----OMEGA BARSAT03000
1-----,15X,-----LAMCA BAR-----,12SAT03010
2X,---EL-----,/)
211 FCRMAT (///)
212 FCRMAT (1,4X,NUMBER OF STATIONS-----,I3/5X,INCREMENTAL LOAD FSAT03030
NUMBER OF MODES-----,F6.3/5X,MAXIMUM NUMBER OF ITERATION-----,I3/5X,MSAT03050
2ACTCR-----,I3/5X,MAXIMUM NUMBER OF ITERATION-----,I3/5X,MSAT03070
3-----,I3/5X,MAXIMUM NUMBER OF ITERATION-----,I3/5X,MSAT03070
4AXIMUM NUMBER OF LOAD FACTOR CHANGES-----,I3/5X,CCNVERGENCE CRITERIASAT03080
5ICN-----,F6.4/)//
213 FCRMAT (1,4X,CHARACTERISTIC SHELL DIMENSION-----,E12.4/5XSAAT03090
1,REFERENCE THICKNESS-----,E12.4/5X,REFERENCE EL-----,E12.4/5XSAAT03100
2ASTICITY-----,E12.4/5X,REFERENCE STRESS-----,E12.4/5XSAAT03110
3-----,E12.4/5X,PCISSON'S RATIO-----,E12.4/5XSAAT03120
4, E12.4/)//
214 FCRMAT (1,110,PLCTS HAVE BEEN REQUESTED FOR THE BELCW LISTED GESAT03140
1CMETRY, STIFFNESS OR LOADING QUANTITIES:/)
215 FCRMAT (1,170,RADIUS,/)
216 FCRMAT (1,170,GAMMA,/)
217 FCRMAT (1,170,GAMMA,/)
218 FCRMAT (1,170,CMEGA-THETA,/)
219 FCRMAT (1,170,DECMEGA-S,/)
22C FCRMAT (1,170,B-STIFFNESS,/)
221 FCRMAT (1,170,B-PRIME,/)
222 FCRMAT (1,170,B-PRIME,/)
SAT02770
SAT02780
SAT02790
SAT02800
SAT02810
SAT02820
SAT02830
SAT02840
SAT02850
SAT02860
SAT02870
SAT02880
SAT02890
SAT02900
SAT02910
SAT02920
SAT02930
SAT02940
SAT02950
SAT02960
SAT02970
SAT02980
SAT02990
SAT03000
SAT03010
SAT03020
SAT03030
SAT03040
SAT03050
SAT03060
SAT03070
SAT03080
SAT03090
SAT03100
SAT03110
SAT03120
SAT03130
SAT03140
SAT03150
SAT03160
SAT03170
SAT03180
SAT03190
SAT03200
SAT03210
SAT03220
SAT03230
SAT03240

```







```

C      NESSES,LOADING (PHYSICAL AND/OR THERMAL), AND INITIAL CCNTIDICNS.*SAT03730
C      IT CCNTROLS PROBLEM SOLUTION PROCEDURE.*SAT03740
C      *****SAT03750
C      INPLICIT LCGICAL*1 ($)
C      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSCIN,MASS
C      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),
C      1ZC(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),IC(55,1),
C      2JC(55,1),PHIXB(1),PHITB(1)
C      CCMMCN /IBL1/ MNMAX
C      CCMMCN /IBL2/ N(99),MNINIT
C      CCMMCN /IBL3/ MO,M1,M2,M3
C      CCMMCN /IBL4/ KMAX,KL
C      CCMMCN /IBL5/ IBCINL,IBCFNL
C      CCMMCN /IBL6/ KLL
C      CCMMCN /IBL7/ MNMAXC,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
C      CCMMCN /IBL8/ LSTEP,ITR
C      CCMMCN /IBL9/ MAXM
C      CCMMCN /IBL10/ IFRREG,NTHMAX
C      CCMMCN /IBL11/ ICORFL,IPASS
C      CCMMCN /IBL12/ KMAX1,KMAX2,ACGNV
C      CCMMCN /IBL13/ ITRMAX,LSMAX
C      CCMMCN /BL1/ A(4,4),BEE(4,4),C(4,4)
C      CCMMCN /BL3/ PR(99),PX(99),PT(99)
C      CCMMCN /BL4/ ZF1M(4,4,99),ZF2M(4,4,99),
C      1 CCMMCN /BL5/ ZF3M(4,4,99),ZF4M(4,4,99),
C      CCMMCN /BL6/ ZF5M(4,4,99),DT(99),DMT(99)
C      CCMMCN /BL7/ MT, APPEARS AS 'EMT' IN SUBRCUTINES INLPOL & FNLPOL
C      CCMMCN /BL8/ SOE,CSE,ALOAD
C      CCMMCN /BL9/ DI,S1
C      CCMMCN /BL10/ R(500),GAM(500),DMT(500)
C      CCMMCN /BL11/ FFS(4,99),ELIS(4),GEES(4,99)
C      CCMMCN /BL12/ PHIX(99),PHIT(99),PHI(99)
C      CCMMCN /BL13/ QMXI(500),PHEE,T0,T2
C      CCMMCN /BL14/ TDLI,TDEL
C      CCMMCN /BL15/ OMEGI(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),
C      1 CCMMCN /BL16/ UNIT(4,4)
C      CCMMCN /BL17/ LAM2,LSD18,LSDIN
C      CCMMCN /BL18/ NU,U1(99),V1(99),W1(99),V2(99),L2(99),U3(99),
C      1 CCMMCN /BL19/ V3(99),W3(99)
C      CCMMCN /BL20/ EPS
C      CCMMCN /BL21/ DEL
C      CCMMCN /BL22/ ELI(4),ELL(4)
C      CCMMCN /BL23/ TH(36)
C      CCMMCN /BL24/ DEQMX(500)
C      CCMMCN /BL25/ JAY(4,4),H(4,4)
C      CCMMCN /BL26/ DL(4,4,99),DG(4,4,99),DF(4,4,99)
C      CCMMCN /BL27/ E(4,4),F(4,4),G(4,4)
C      CCMMCN /BL28/
C      SAT04050
C      SAT04060
C      SAT04070
C      SAT04080
C      SAT04090
C      SAT04100
C      SAT04110
C      SAT04120
C      SAT04130
C      SAT04140
C      SAT04150
C      SAT04160
C      SAT04170
C      SAT04180
C      SAT04200

```







SAT04680  
SAT04690  
SAT04700  
SAT04710  
SAT04720  
SAT04730  
SAT04740  
SAT04750  
SAT04760  
SAT04770  
SAT04780  
SAT04790  
SAT04800  
SAT04810  
SAT04820  
SAT04830  
SAT04840

SAT04850  
SAT04860  
SAT04870  
SAT04880  
SAT04890  
SAT04900  
SAT04910  
SAT04920  
SAT04930  
SAT04940  
SAT04950  
SAT04960  
SAT04970  
SAT04980  
SAT04990  
SAT05000  
SAT05010  
SAT05020  
SAT05030  
SAT05040  
SAT05050  
SAT05060  
SAT05070  
SAT05080  
SAT05090  
SAT05100  
SAT05110  
SAT05120  
SAT05130

```

CC 99 I=1,4
CC 99 J=1,4
KKLM=J/4+1
CMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)
CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)
LAM=TKN/CHAR
LCE=SIGC/ELAST
CCE=.5*SOE
CI=1.0-NU
SI=1.0+NU
LAM2=LAM**2
IF(NDIMEN.LT.1) GC TO 228
SIGC=1.0
ELAST=1.0
TKA=1.0
CFAR=1.0
CC 230 N=1,MAXM
PFX(M)=C.0
PFT(M)=C.0
N(N)=0.0
PX(M)=0.0
PT(M)=0.0
PR(M)=0.0
TT(M)=0.0
NT(M)=0.0
LT(M)=0.0
CNT(M)=C.0
MAXD(M)=0
MAXS(M)=0
MAXSY(M)=0
CALL GECM
1 ICFCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(ICEOMS)
  +IABS(IRADII)+IABS(IRADII)
  ICFCK2=IAES(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(ICDSTF)
  IF (.NOT.$PLOTS) GC TO 1001
CC 2 K=1,KMAX
XSTATN(K)=FLOAT(K)
2 IF (ICFCK1.EQ.0) GC TO 1001
CC 1 K=1,KMAX
XRADII(K)=R(K)*CHAR
YGAMMA(K)=GAM(K)/CHAR
YCMEGS(K)=CMXI(K)/CHAR
YCMEGT(K)=CMT(K)/CHAR
YLECMS(K)=DEOMX(K)/(CHAR*CHAR)
1 CCNTINUE
1001 CCNTINUE
CC 86 K=1,KMAX
66 MASS(K)=0.

```



```

WRITE(6,802)
CC 578 K=1,KMAX
RKK=R(K)*CHAR
CMXIK=CMXI(K)/CHAR
GAMK=GAM(K)/CHAR
CMTK=CMT(K)/CHAR
CECMXK=CECMX(K)/(CHAR*CFAR)
WRITE(6,803) K,RKK,GAMK,CMXIK,OMTK,DEGMXK
978
803
N1=0
N2=C
N3=C
AEN=CHAR/SIGQ/TKN
ZN=SIGQ*TKN
WRITE(6,112)
CC 888 K=1,KMAX
CALL BCE(K,B,DB,D,DD)
EST=ELAST*TKN
ZST=ELAST*TKN**3
B=B*BST
C=C*ZST
CB=DB/CFAR*BST
CC=CC/CFAR*ZST
WRITE(6,71) K,B,D,CB,DD
IF (.NOT. $PLCTS.OR.(ICCHK2.EQ.0)) GO TO 888
YESIF(K)=B
YECSTIF(K)=CB
YECSTIF(K)=CD
CCNTINLE
CALL PLCAD(1,Z)
CALL TLCAD(1,Z)
CC 889 N=1,MNMAX
WRITE(6,113) N(M)
WRITE(6,114)
ICCHK3=IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)+IABS(IMT)
1
DC 890 K=1,KMAX
CALL PLCAD(K,Z)
CALL TLCAD(K,Z)
PFM=PR(M)/ABN
PTM=PT(M)/ABN
FXM=PX(M)/ABN
TMM=TT(M)*ZN
ENTM=MT(M)/CHAR*ZN
LTM=DT(M)/CHAR*ZN
CMTM=DMT(M)*ZN*TKN/(CHAR*CHAR)
WRITE(6,115) K,PRM,PXM,PTM,TTM,ENTM,DTM,CMTM
SAT05140
SAT05150
SAT05160
SAT05170
SAT05180
SAT05190
SAT05200
SAT05210
SAT05220
SAT05230
SAT05240
SAT05250
SAT05260
SAT05270
SAT05280
SAT05290
SAT05300
SAT05310
SAT05320
SAT05330
SAT05340
SAT05350
SAT05360
SAT05370
SAT05380
SAT05390
SAT05400
SAT05410
SAT05420
SAT05430
SAT05440
SAT05450
SAT05460
SAT05470
SAT05480
SAT05490
SAT05500
SAT05510
SAT05520
SAT05530
SAT05540
SAT05550
SAT05560
SAT05570
SAT05580
SAT05590
SAT05600
SAT05610

```



```

IF (.NCT.$PLCTS.GR.(ICHCK3.EQ.0)) GO TO 890
YPR(K)=PRM
YFS(K)=PXN
YPT(K)=PTM
YTT(K)=TTM
YMT(K)=ETM
YCTI(K)=DTM
YCMT(K)=DMTM
85C CCNTINUE
IF (M.EQ.1) ICHCK3=ICHCK1+ICHCK2+ICHCK3
IF ({PLCTS.AND.({ICFK3.GT.0}) CALL PLOT1(M)
85S CCNTINUE
CELSQ=DEL**2
TCLI=.5/DEL
TCEL=2.C*DEL
MNNIT=1
MNNAXC=MNNAX
CC 20 I=1,4
CC 20 J=1,4
LNIT(I,J)=0.0
IF(I.EQ.J) UNIT(I,J)=1.0
2C NMAX=MXXM*KMAX2
CC 22 K=1,NMAX
CC 22 I=1,4
ZCCT(I,K)=0.0
ZC(I,K)=0.0
Z2(I,K)=0.0
Z3(I,K)=0.0
Z(I,K)=0.0
22 ALCAD=DELCAD
CALL IMPERF (PHIXB,PHITB)
CALL PMATRX (P,X,ZC,Z2,Z3,DEE,DST)
LSTEP=1
LCHANG=0
ITR=1
ICCRFL=0
IF(MNMAX.EQ.MAXM) ICORFL=1
IPASS=0
CALL XANDZ (P,DEE,DST,X,Z,ZC,Z2,Z3,ZCOT,IS,JS,ID,JD,PTIXB,PHITB)
4CC IF(ITRMAX.EQ.1) GC TO 50
MNNAXC=MNNAX
IF(IPASS.LT.2) CALL MDES (IS,JS,IC,JD,P,X,ZC,Z2,Z3,DEE,DST)
IF(INCONV.EQ.1).AND.(ITR.GT.1)) GC TO 50
IF(ITR.LT.ITRMAX) GC TO 23
IF(LCHANG.LT.LCHMAX) GO TO 30
WRITE(6,220) NO
GC TO 500
5C FL=LSTEP

```

SAT05620  
 SAT05630  
 SAT05640  
 SAT05650  
 SAT05660  
 SAT05670  
 SAT05680  
 SAT05690  
 SAT05700  
 SAT05710  
 SAT05720  
 SAT05730  
 SAT05740  
 SAT05750  
 SAT05760  
 SAT05770  
 SAT05780  
 SAT05790  
 SAT05800  
 SAT05810  
 SAT05820  
 SAT05830  
 SAT05840  
 SAT05850  
 SAT05860  
 SAT05870  
 SAT05880  
 SAT05890  
 SAT05900  
 SAT05910  
 SAT05920  
 SAT05930  
 SAT05940  
 SAT05950  
 SAT05960  
 SAT05970  
 SAT05980  
 SAT05990  
 SAT06000  
 SAT06010  
 SAT06020  
 SAT06040  
 SAT06050  
 SAT06060  
 SAT06070  
 SAT06080  
 SAT06090



```

FI=IPRINT
LI=LSTEP/IPRINT
LI=LI
FT=FLI-FL/FI
1 IF(FT.EC.0.) CALL CPUTUT(IMCDE,P,CEE,DST,X,2,ZC,Z2,Z3,ZCCT,IS,JS,
  IF(JD,P,IXB,PHITB)
  IF(LSTEP.EC.1) ITR=1
  IF(LSTEP.EC.1) ITRPR=1
  IF(ITR.GT.ITRPR) ITRPR=ITR
  IF(LSTEP.GE.LSMAX) GO TO 360
CC 61 MN=1,MNMAX
CC 61 K=1,KMAX2
IK=K+(MN-1)*KMAX2
CC 61 I=1,4
ZN=2.0*Z(I,IK)-ZC(I,IK)
ZC(I,IK)=Z(I,IK)
Z(I,IK)=ZN
61 IF(LSTEP.GE.LSMAX) GO TO 360
62 ALCAC=ALQAC+DELOAD
LSTEP=LSTEP+1
ITR=1
CC TO 400
WRITE(6,221) NO
36C GC TO 500
23 ITR=ITR+1
3C GC TO 400
310 IF(LSTEP-1) 310,310,320
GC TO 500
32C WRITE(6,222)
LCFANG=LCHANG+1
LSTEP=LSTEP-1
ALCAC=ALQAC-DELOAD
CC 32 MN=1,MNMAX
CC 32 K=1,KMAX2
IK=K+(MN-1)*KMAX2
CC 32 I=1,4
Z(I,IK)=ZC(I,IK)
32 GC TO 62
C*****
71 FCRMAT(20X,13,4X,4E20.6)
112 FCRMAT(///17X,12F 20H STATION 20H B STIFFNESS 20H C ST
112 1IFFNESS 20H B PRIME 20H D PRIME //)
113 FCRMAT(///25X,44+PRESSURE AND TEMPERATURE CCEFFICIENTS FOR N=13,8H
114 1 FLOW//)
114 FCRMAT(5X,7HSTATION,3X,15H PR 15H MT 15H PX 15H DTT 15H
1 PT
SAT06100
SAT06110
SAT06120
SAT06130
SAT06140
SAT06150
SAT06160
SAT06170
SAT06180
SAT06190
SAT06200
SAT06210
SAT06220
SAT06230
SAT06240
SAT06250
SAT06260
SAT06270
SAT06280
SAT06290
SAT06300
SAT06310
SAT06320
SAT06330
SAT06340
SAT06350
SAT06360
SAT06370
SAT06380
SAT06390
SAT06400
SAT06410
SAT06420
SAT06430
SAT06440
SAT06450
SAT06460
SAT06470
SAT06480
SAT06490
SAT06500
SAT06510
SAT06520
SAT06530
SAT06540
SAT06550
SAT06560
SAT06570

```



```

25+ DMT //)
115 FCRMAT(6X,I3,7X,7E15.4) THE MAXIMUM NUMBER CF LOAC CHANGES HAVE BEEN
220 FCRMAT(IH1,80H) THE MAXIMUM NUMBER CF LOAC STEPS HAVE BEEN
221 1EN MADE. END PROBLEM NUMBERI4) THE MAXIMUM NUMBER CF LOAC STEPS HAVE BEEN
222 1 FCRMAT(IH1,79H) THE MAXIMUM NUMBERI4) THE MAXIMUM NUMBER CF LOAC STEPS HAVE BEEN
223 1 TAKEN. END PROBLEM NUMBERI4) THE MAXIMUM NUMBERI4) THE MAXIMUM NUMBER CF LOAC STEPS HAVE BEEN
224 1 FCRMAT(IH1,119H) THE SOLUTION DID NOT CONVERGE WITHIN THE MAXIMUM NUMBER OF
225 1 MAXIMUM NUMBER OF ITERATIONS. THE LOAD FACTOR HAS BEEN DIVIDED BY
226 1 FCRMAT(IH1,69H) THE SOLUTION DID NOT CONVERGE FOR THE FIRST
227 1 LOAD INCREMENT.//11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR TRY A
228 1 SMALLER VALUE FOR DELCAD.) 16H RACILS 16H GAMMA
802 1 FCRMAT(IH1,17X,15H STATION 16H CMEGA THETA16H DECMEGA S //)
803 1 FCRMAT(20X,I3,9X,5E16.4)
888 FCRMAT (00,T20, EXECUTING IN SUBROUTINE "STATIC")
500 RETURN
END
SUBROUTINE DYNAMC (P,CEE,DST,X,Z,ZC,Z2,Z3,ZCCT,IS,JS,ID,JD,PHIXB,
1PHITB)
C*****
C THIS SUBROUTINE IS ONE OF THE MAJOR CONTRCLLING SUBROUTINES FCR
C ALL DYNAMIC ANALYSIS PROBLEMS. IT OPERATES IN A FASHION SIMILAR
C TO SUBROUTINE STATIC.
C*****
C IMPLICIT LOGICAL*1 ($)
REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
DIMENSICN P(4,4,1),CEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),
1ZC(4,1),Z2(4,1),Z3(4,1),ZCCT(4,1),JS(99,1),ID(99,1),
2JC(99,1),PHIXB(1),PHITB(1)
CCMMCN /IBL1/ MNMAX
CCMMCN /IBL2/ N(99),MNINIT
CCMMCN /IBL3/ MO,M1,M2,M3
CCMMCN /IBL4/ KMAX,KL
CCMMCN /IBL5/ IBCINL,IBCFNL
CCMMCN /IBL6/ KLL
CCMMCN /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
CCMMCN /IBL8/ LSTEP,ITR
CCMMCN /IBL9/ MAXM
CCMMCN /IBL10/ IFREQ,NTHMAX
CCMMCN /IBL11/ ICCRFL,IPASS
CCMMCN /IBL12/ KMAX1,KMAX2,ACONV
CCMMCN /IBL13/ ITRMAX,LSMAX
CCMMCN /BL1/ A(4,4),BEE(4,4),C(4,4)
CCMMCN /BL2/ PR(99),PX(99),PT(99)
CCMMCN /BL3/ ZFIM(4,4,99),ZF2M(4,4,99),
CCMMCN /BL4/ ZF3M(4,4,99),ZF4M(4,4,99)
1

```



CCMMCN	/BL5/	TT(99),MT(99),DT(99),DMT(99) ,MT, APPEARS AS 'EMT' IN SUBRCUTINES INLPCL & FNLPOL	SAT07060
CCMMCN	/BL6/	SOE, QSE, ALOAD	SAT07070
CCMMCN	/BL7/	D1, S1	SAT07080
CCMMCN	/BL8/	R(500), GAM(500), QMT(500)	SAT07090
CCMMCN	/BL9/	FFS(4,99), ELIS(4), CEES(4,99)	SAT07100
CCMMCN	/BL10/	PHIX(99), PHIT(99), PFI(99)	
CCMMCN	/BL11/	OMXI(500), PHEE, IO, I2	
CCMMCN	/BL12/	TDLI, TDEL	
CCMMCN	/BL13/	OMEGI(4,4), CAPLI(4,4), OMEGL(4,4), CAPLL(4,4),	
1		UNIT(4,4)	
CCMMCN	/BL14/	LAM2, LSD18, LSC1N	SAT07120
CCMMCN	/BL15/	NU, U1(99), V1(99), W1(99), V2(99), L2(99), W2(99), U3(99),	SAT07130
1		V3(99), W3(99)	SAT07140
CCMMCN	/BL16/	EPS	SAT07150
CCMMCN	/BL17/	DEL	SAT07160
CCMMCN	/BL18/	ELI(4), ELL(4)	SAT07170
CCMMCN	/BL19/	TH(36)	SAT07180
CCMMCN	/BL20/	DEOMX(500)	SAT07190
CCMMCN	/BL23/	JAY(4,4), F(4,4)	SAT07200
CCMMCN	/BL24/	DL(4,4,99), DG(4,4,99), DF(4,4,99)	SAT07210
CCMMCN	/BL25/	E(4,4), F(4,4), G(4,4)	SAT07220
CCMMCN	/BL27/	BX3(99), BT3(99), BX12(99), BE3(99)	SAT07230
CCMMCN	/BL28/	EXX3(99), ETX3(99), ETX13(99), EX3(99), ET3(99)	SAT07240
CCMMCN	/BL29/	BX1(99), BT1(99), BX11(99), BE1(99), B12(99),	SAT07250
1		BX12(99), BE2(99)	
CCMMCN	/BL30/	EXX1(99), ETX1(99), ETX1(99), ETX1(99), EXX2(99),	
1		ETX2(99), ETX2(99), EX2(99), ET2(99)	
CCMMCN	/BL31/	DELSQ, EX11(99)	SAT07270
CCMMCN	/BL32/	TKN, ELAST, CHAR, SIGC	SAT07280
CCMMCN	/BL100/	TEEG, \$DYNMC	SAT07290
CCMMCN	/BL101/	DELSQ	SAT07300
CCMMCN	/BL102/	DELOAD	SAT07310
CCMMCN	/BL103/	MASS(500)	SAT07320
CCMMCN	/BL110/	TX(99), TTH(99), TXI(99), MX(99), MTH(99), MXT(99),	SAT07330
1		QS(99)	SAT07340
CCMMCN	/BL111/	ABZ, ABZC, ABZN, ABZ3, DC2	SAT07350
CCMMCN	/BLPFS/	IRACII, IGAMMA, ICMEGS, IOMEGT, ICEOMS, IBSTIF, IDSTIF,	SAT07360
CCMMCN	/BLFLUT/	IBBSTF, IDBSTF, IPR, IPS, IPT, IIT, IMT, ICIT, ICM, IAS,	SAT07370
		INTH, INSTH, IQS, IQS, IMH, IMSH, IU, IV, IW, IFHIS,	SAT07380
		IPHIT, IPHI, \$PLOTS, \$MODAL	SAT07390
		XRADII(200), YGAMMA(200), YCMEGS(200), YCMEGT(200),	SAT07400
CCMMCN	/BLPLTI/	YDECM(200), YBSTIF(200), YPT(200), YPT(200),	SAT07410
		YDDSTF(200), YPR(200), YPS(200), YPT(200), YTT(200),	SAT07420
		YMT(200), YDT(200), YDMT(200), YNS(200), YNT(200),	
		YNSTH(200), YQS(200), YMS(200), YMTH(200), YMSTH(200),	
		YL(200), YV(200), YW(200), YPHIS(200), YPFI(200),	
1			SAT07430
2			SAT07440
3			SAT07450
			SAT07460
			SAT07470
1			SAT07480
2			SAT07490
3			SAT07500
4			SAT07510
5			SAT07520



```

6  CCOMMON /BLDATA/ YPHI(200),XSTATN(200)
   DIMENSION SIGT(2),TITLE,NC,MODE,NCIMEN,IPRINT,LCHMAX,IC
   DIMENSION VBAR(99),SIGC(2),TITLE(18)
   C*****
   WRITE(6,8888)
   CELSD=DELCAD*DELCAD
   KLL=KMAX-1
   KMAX1=KMAX+1
   KMAX2=KMAX+2
   AK=KL
   SIGT(1)=SIGC*TKNST
   SIGT(2)=SIGC/ELAST
   SIGC(1)=SIGC*CHAR/ELAST
   SIGC(2)=SIGC*TKN**3/CHAR
   IF (IBCINL.LT.0) GO TO 14
   DO 58 I=1,4
   DO 58 J=1,4
   KKLM=J/4+1
   OMEGL(I,J)=OMEG1(I,J)*SIGT(KKLM)
   CAPLL(I,J)=CAPLI(I,J)*SIGC(KKLM)
   IF (IBCFNL.LT.0) GO TO 17
   DO 59 I=1,4
   DO 59 J=1,4
   KKLM=J/4+1
   OMEGL(I,J)=OMEG1(I,J)*SIGT(KKLM)
   CAPLL(I,J)=CAPLI(I,J)*SIGC(KKLM)
   LAM=TKN/CHAR
   SCE=.5*SCE
   DI=1.0-NU
   SI=1.0+NU
   LAM2=LAM**2
   IF (NDIMEN.LT.1) GO TO 228
   SIGC=1.0
   ELAST=1.0
   TKN=1.0
   CC 230 N=1,MAXM
   PFX(M)=C.0
   PFT(M)=0.0
   N(M)=0.0
   PX(M)=0.0
   PT(M)=0.0
   PR(M)=0.0
   TT(M)=0.0
   NT(M)=0.0
   SAT07530
   SAT07540
   SAT07550
   SAT07560
   SAT07570
   SAT07580
   SAT07590
   SAT07600
   SAT07610
   SAT07620
   SAT07630
   SAT07640
   SAT07650
   SAT07660
   SAT07670
   SAT07680
   SAT07690
   SAT07700
   SAT07710
   SAT07720
   SAT07730
   SAT07740
   SAT07750
   SAT07760
   SAT07770
   SAT07780
   SAT07790
   SAT07800
   SAT07810
   SAT07820
   SAT07830
   SAT07840
   SAT07850
   SAT07860
   SAT07870
   SAT07880
   SAT07890
   SAT07900
   SAT07910
   SAT07920

   SAT07930
   SAT07940
   SAT07950
   SAT07960
   SAT07970
   SAT07980

```



SAT07590  
SAT08000  
SAT08010  
SAT08020  
SAT08030  
SAT08040  
SAT08050  
SAT08060  
SAT08070  
SAT08080  
SAT08090  
SAT08100  
SAT08110  
SAT08120  
SAT08130  
SAT08140  
SAT08150  
SAT08160  
SAT08170  
SAT08180  
SAT08190  
SAT08200  
SAT08210  
SAT08220  
SAT08230  
SAT08240  
SAT08250  
SAT08260  
SAT08270  
SAT08280  
SAT08290  
SAT08300  
SAT08310  
SAT08320  
SAT08330  
SAT08340  
SAT08350  
SAT08360  
SAT08370  
SAT08380  
SAT08390  
SAT08400  
SAT08410  
SAT08420  
SAT08430  
SAT08440  
SAT08450  
SAT08460

```

CT(M)=0.0
DMT(M)=C.0
MAXC(M)=0
MAXS(M)=0
MAXSY(M)=0
CALL GECM
ICFCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(ICEOMS)
1 ICFCK2=IABS(IRADII)+IABS(ISTIF)+IABS(ISTIF)+IABS(ISTIF)
IF (.NOT. $PLOTS) GO TO 1001
IC 2 K=1,KMAX
XSTATN(K)=FLQAT(K)
2 IF (ICHECK1.EQ.0) GC TO 1001
CC 1 K=1,KMAX
XRADII(K)=R(K)*CHAR
YGAMMA(K)=GAM(K)/CHAR
YCMEGS(K)=OMXI(K)/CHAR
YCMEGT(K)=GMT(K)/CHAR
YCECMS(K)=DEOMX(K)/(CHAR*CHAR)
CCNTINUE
10C1 WRITE(6,810)
8C4 TEEC=TEEC
IF (NDIMEN.EQ.1) TEEC=1.0
CC 579 K=1,KMAX
RKK=R(K)*CHAR
CMXIK=CMXI(K)/CHAR
GAMK=GAM(K)/CHAR
GMTK=GMT(K)/CHAR
DECMXK=DEOMX(K)/(CHAR*CHAR)
AMSS=MASS(K)*TEED*2*ELAST*TKN/CHAR**2
57C WRITE(6,813) K,RKK,GAMK,CMXIK,GMTK,DEOMXK,AMSS
8C5 MC=0
M1=0
M2=0
M3=0
AEN=CHAR/SIGO/TKN
ZN=SIGC*TKN
WRITE(6,112)
CC 888 K=1,KMAX
CALL BDE(K,B,DB,D,CC)
EST=ELAST*TKN**3
ZST=ELAST*TKN**3
B=B*BST
C=C*ZST
CE=CB/CHAR*BST
CC=DD/CHAR*ZST
WRITE (6,71) K,B,C,CB,DC

```



IF (.NOT. \$PLOTS.OR.(ICHECK2.EQ.0)) GC TO 888

SAT08470  
SAT08480  
SAT08490  
SAT08500  
SAT08510  
SAT08520  
SAT08530  
SAT08540  
SAT08550  
SAT08560  
SAT08570  
SAT08580  
SAT08590  
SAT08600  
SAT08610  
SAT08620  
SAT08630  
SAT08640  
SAT08650  
SAT08660  
SAT08670  
SAT08680  
SAT08690  
SAT08700  
SAT08710  
SAT08720  
SAT08730  
SAT08740  
SAT08750  
SAT08760  
SAT08770  
SAT08780  
SAT08790  
SAT08800  
SAT08810  
SAT08820  
SAT08830  
SAT08840  
SAT08850  
SAT08860  
SAT08870  
SAT08880  
SAT08890  
SAT08900  
SAT08910  
SAT08920  
SAT08930  
SAT08940

```

888 IF (.NOT. $PLOTS.OR.(ICHECK2.EQ.0)) GC TO 888
    YESIF(K)=B
    YCSTIF(K)=C
    YRBSIF(K)=DB
    YCLSTIF(K)=CC
    CCNTINUE
    CALL FLCAD(1,Z)
    CALL TLCAD(1,Z)
    CELSQ=DEL#2
    TCLLI=.5/DEL
    TCEL=2.C*DEL
    MNINIT=1
    MNMAXC=MNMAX
    DC 20 I=1,4
    DC 20 J=1,4
    UNIT(I,J)=0.0
    IF(I.EQ.J) UNIT(I,J)=1.0
    NMAX=MAXM*KMAX2
    CC 22 K=1,MNMAX
    DC 22 I=1,4
    ZCCT(I,K)=0.0
    ZC(I,K)=0.0
    Z2(I,K)=0.0
    Z3(I,K)=0.0
    Z(I,K)=0.0
    IF(IC.EC.0) GO TO 834
    CALL INITL(Z,Z0,Z2,Z3,ZCOT)
    ACC=CHAR*SIG/ELAST
    ACM=SIGC*TKN**3/CHAR
    DC 830 M=1,MNMAX
    MN=(M-1)*KMAX2
    WRITE(6,126) N(M)
    WRITE(6,127)
    DO 831 K=2,KMAX1
    MK=K+MM
    TL=ACQ*ZO(1,MK)
    TV=ACQ*ZO(2,MK)
    TW=ACQ*ZO(3,MK)
    TN=ACM*ZC(4,MK)
    KK=K-1
    WRITE(6,71) KK,TU,TV,TW,TM
    WRITE(6,129)
    CC 830 K=2,KMAX1
    ACC=CHAR*SIG/(ELAST*TEEO)
    ACM=SIGC*TKN**3/(CHAR*TEEO)
    MK=K+MM
    TL=ACQ*ZDOT(1,MK)
    TV=ACQ*ZDOT(2,MK)

```



```

      TW=ACD*ZDOT(3,MK)
      TM=AMD*ZDOT(4,MK)
      KK=K-1
      WRITE(6,71) KK,TU,TV,TH,TM
      CC 830 I=1,4
      Z(I,MK)=Z(I,MK)+ZCCT(I,MK)*DELCAC
      Z2(I,MK)=Z(I,MK)-ZDOT(I,MK)*DELCAC
      Z3(I,MK)=Z(I,MK)-2.*ZDOT(I,MK)*DELCAD
      CCNTINUE
      ALCAD=1.0
      CALL IMPERF (PHIXB,PHITB)
      CALL FMATRIX (P,X,ZC,Z2,Z3,DEE,DST)
      LSTEP=1
      LCFANG=C
      ITR=1
      ICCRFL=0
      IF(MNMAX.EQ.MAXM) ICORFL=1
      IPASS=0
      ITTEST=0
400  CALL XANDZ (P,DEE,DST,X,Z,ZC,Z2,Z3,ZDOT,IS,JS,IC,JD,JC,PTIXB,PHITB)
      IF(ITRMAX.EQ.1) GO TO 50
      MNMAX=MNMAX
      IF(IPASS.LT.2) CALL MCDES (IS,JS,IC,JD,P,X,ZC,Z2,Z3,CEE,CST)
      IF(NCCNV.EQ.1) GO TO 50
      IF(ITR.LT.ITRMAX) GC TO 23
      GC TO 365
50  FL=LSTEP
      FI=IPRINT
      LI=LSTEP/IPRINT
      FLI=LI
      FT=FLI-FL/FI
      IF(FT.EQ.0) CALL OUTPUT(IMODE,P,CEE,DST,X,Z,ZC,Z2,Z3,ZCOT,IS,JS,
1  IL,JD,PTIXB,PHITB)
      IF(LSTEP.GE.LSMAX) GC TC 360
      CC 65 MN=1,MNMAX0
      DC 65 K=1,KMAX2
      IK=K+(MN-1)*KMAX2
      DN 65 I=1,4
      ZN=3.0*(Z(I,IK)-ZC(I,IK))+Z2(I,IK)
      Z3(I,IK)=Z2(I,IK)
      Z2(I,IK)=ZC(I,IK)
      ZC(I,IK)=ZN
      Z(I,IK)=ZN
      ALCAD=1.0
      LSTEP=LSTEP+1
      ITR=1
      GC TO 400
23  ITR=ITR+1

```

SAT08950  
 SAT08960  
 SAT08970  
 SAT08980  
 SAT08990  
 SAT09000  
 SAT09010  
 SAT09020  
 SAT09030  
 SAT09040  
 SAT09050  
 SAT09060  
 SAT09070  
 SAT09080  
 SAT09090  
 SAT09100  
 SAT09110  
 SAT09120  
  
 SAT09130  
 SAT09140  
 SAT09150  
  
 SAT09170  
 SAT09180  
 SAT09190  
 SAT09200  
 SAT09210  
 SAT09220  
 SAT09230  
 SAT09240  
 SAT09250  
 SAT09260  
 SAT09300  
 SAT09310  
 SAT09320  
 SAT09330  
 SAT09340  
 SAT09350  
 SAT09360  
 SAT09370  
 SAT09380  
 SAT09390  
 SAT09400  
 SAT09410  
 SAT09420  
 SAT09430  
 SAT09440



```

36C GC TO 400
    WRITE(6,271) (AVB(M),M=1,MAXM)
    WRITE(6,188) ITRPR
    GC TO 500
365 IF(LSTEP.EQ.1) GO TC 367
    WRITE(6,266) ITRMAX,LSTEP,NC
    WRITE(6,188) (AVB(M),M=1,MAXM)
    GC TO 500
367 WRITE(6,273)
    GC TO 500
C*****
71 FCRMAT(20X,13,4X,4E20,6)
112 FCRMAT(///17X,12H STATION 20F 20H B STIFFNESS 20F 20H D PRIME 20F 20H C STS
126 FCRMAT(///5X,29H THE INITIAL CONCTIONS FOR N=13,8H FOLLOW//)
127 FCRMAT(19X,7H STATION,3X,20H U M S U DOT M S DOT V DOT
129 FCRMAT(///19X,7H STATION,3X,20H W DOT M S DOT M S DOT V DOT
188 FCRMAT(/// THE MAXIMUM VBAR 2CH EACH MCCE IS,10E11,4)
189 FCRMAT(/// THE MAXIMUM NUMBER CF ITERATIONS TAKEN IS,13)
266 FCRMAT(1H,35H THE SOLUTION DID NOT CONVERGE IN 13,24H ITERATIONS
271 FCRMAT(1H,75H AT TIME STEP 5,21H. END PROBLEM NUMBER 14,1F.)
273 FCRMAT(1H,69H TAKEN. END PROBLEM NUMBER 14)
1T TIME INCREMENT.11X,71H LOCK FOR AN ERRCR IN THE INPUT DATA, CR T
2RY A SMALLER VALUE FOR DELOAD.)
81C FCRMAT(1H,5X,15H STATION 16H RADIUS 16H GAMMA
16H CMEGA THETA 16H CMEGA S 16H
813 FCRMAT(8X,13,9X,6E16,4)
888 FCRMAT(.,10,.,EXECUTING IN SUBROUTINE "DYNAMIC")
5CC RETURN
C*****
SUBROUTINE PLOTIT(X,Y,NN,MODCUR)
C*****
C THIS SUBROUTINE AND THE THREE THAT FOLLOW IT COMPRISE THE SELF-
C CONTAINED PLOTTING CAPABILITY OF PROGRAM SATANS. THEY RECEIVE
C DATA TO BE PLOTTED, ROUND IT, SCALE IT, AND DRAW IS CN THE HIGH-
C SPEED LINE PRINTER.
C*****
C DIMENSION X(1),Y(1), RANGE(4)
C EQUIVALENCE (RANGE(1),XMAX), (RANGE(2),XMIN), (RANGE(3),YMAX),
1 KN=IABS(NN)
IF(MODCUR.GT.1) GC TO 5

```







```

C** RCUND MAXIMUM TO NEXT HIGHEST 2 SIG FIGS
C** XMAX=AMAX1(0.,XMAX)
C** CALL RCUND(XMAX,IMX,FMX)
C** IMX=IMX-1
C** FMX=FMX*10.
3 XMAX=FMX*10.
C** IF(XMAX.GE.XMAX) GC TO 2
C** FMX=FMX+1.
C** FMX=FMX
C** FMX=IMM
C** GC TO 3
C** RCUND MINIMUM TO NEXT LOWEST 2 SIG FIGS
C** XMIN=AMIN1(0.,XMIN)
2 CALL RCUND(XMIN,IMN,FMN)
C** IMN=IMN-1
C** FMN=FMN*10.
14 XMIN=FMN*10.
C** IF(XMIN.GE.XMN) GC TO 11
C** FMN=FMN-1.
C** FMN=FMN
C** FMN=IMM
C** GC TO 14
C** RCUND MAX & MIN TC 1. OR .1 IF RANGE LARGE
C** XSC=XMX-XMN
11 IN=C
C** SA=1.
5 IF(XSC/CIV.LE.SM) GC TO 12
C** IF(ABS(XMN).LT.SM.AND.ABS(XMN).GT.0.) XMN=SIGN(SM,XMN)
C** IF(ABS(XMX).LT.SM.AND.ABS(XMX).GT.0.) XMX=SIGN(SM,XMX)
12 IF(IM.GT.0) GO TO 19
C** SM=1
C** IN=IM+1
C** GC TO 9
C** RCUND RANGE (MAX-MIN) TC 2 SIG FIGS
C** XSC=XMX-XMN
15 CALL RCUND(XSC,ISIC,FACTX)
C** FINE FACTOR WHICH IS MULTIPLE OF IDIV
C** FACTX=FACTX*10.
C** CFAC=FACTX

```



```

IS10=IS10-1
IFX=FACTX
FACTX=IFX
2C IF(MCC(IFX, IDIV).EQ.0.AND.FACTX.GE.OFAC) GO TC 10
IFX=IFX+1
FACTX=IFX
GC TO 20
1C IF(IDIV.GT.4) GO TC 15
C *****
C *****
C *****
C *****
FFX=ABS(FACTX/10.)
IF(FFX.GT.8.AND.FFX.LT.10.) FFX=10.
IF(FACT X.LT.0.) FFX=-10.
FACTX=FFX*10.
15 XSC=FACTX*10.**IS10
C *****
C *****
C *****
C *****
CCOMPUTE NEW MAX & MIN FROM ROUNDED SCALE
C *****
C *****
IF(XM N*XM X.NE.0.) GO TO 4
IF(XM N.LT.0.) XMIN=-XSC
IF(XM X.GT.C.) XMAX=XSC
RETURN
4 XMAX=XSC+XMN
XMIN=XMN
RETURN
END
C *****
C *****
C *****
C *****
SLBROUTINE ROUNDA(ANUM,IS,FACT)
C *****
C *****
C *****
C *****
EXPRESS ANUM IN SCIENTIFIC NOTATION WHERE
ANUM=FACT*10.**IS WHERE FACT IS BETWEEN 1. AND 5.9
C *****
C *****
C *****
C *****
IF(ANUM.EQ.0.) GO TC 15
ANUM=ANUM
BNUM=-BNUM
IF(BNUM.LT.0.) BNUM=43429448
IS=ALCG(BNUM).**IS
FACT=BNUM/10.**IS
C *****
C *****
C *****
C *****
FIND POWER CF 10
C *****
C *****
C *****
ICC=-3
R2=0
DO 10 I1=1,5
ICC=ICC+1
R1=R2
R2=10.**(ICC+1)
IF(FACT.GE.R1.AND.FACT.LT.R2) GC TO 8

```



```

1C CCNTINUE
1E FACT=FACT*10.**(-IDD)
  IS=IS+IDD
C *****
C RCLND MANTISSA TO 2 SIG FIGS
C *****
  IFAC=FACT*10.+0.05
  FACT=IFAC
  FACT=FACT/10.
  IF(FACT.LT.10.) GC TO 20
C *****
C SET TO 1 IF LESS THAN 10.
C *****
  FACT=1.
  IS=IS+1
C *****
C IF INPUT NEGATIVE, SET MANTISSA NEGATIVE
C *****
  ZC IF(ANUM.LT.0.) FACT=-FACT
  RETURN
C *****
C SET TO C. IF 0.
C *****
  IF FACT=0.
  IS=C
  RETURN
C *****
C *****
  SUBROUTINE DRAWIT(X,Y,NCATA,RANGE,KKZ,MOCCUR)
  DIMENSION GRID(61,81),XSCALE(5),YSCALE(7)
  DIMENSION X(1),Y(1),RANGE(4)
  INTEGER*2 GRID,BLANK,CCT,XCHAR(4)/1H+,1H.,1H+,1H+/
  DATA DOT,BLANK/Z4B40,Z4040/
  KDATA=NCATA*KKZ
  IF(MOCCUR.GT.1) GC TO 444
C *****
C GRID IS THE MATRIX USED TO PLOT THE POINTS
C *****
  IERR=0
  XMAX=RANGE(1)
  XMIN=RANGE(2)
  YMAX=RANGE(3)
  YMIN=RANGE(4)
C *****
C CHECKING X AND Y PCINTS AND PLOTTING THOSE CUT OF RANGE
C *****
C AT THE MARGIN
C *****
  CC 30 I=1,KDATA,KKZ

```

SAT111370  
 SAT111380  
 SAT111390  
 SAT111400  
 SAT111410  
 SAT111420  
 SAT111430  
 SAT111440  
 SAT111450  
 SAT111460  
 SAT111470  
 SAT111480  
 SAT111490  
 SAT111500  
 SAT111510  
 SAT111520  
 SAT111530  
 SAT111540  
 SAT111550  
 SAT111560  
 SAT111570  
 SAT111580  
 SAT111590  
 SAT111600  
 SAT111610  
 SAT111620  
 SAT111630  
 SAT111640  
 SAT111650  
 SAT111660  
 SAT111670  
 SAT111680  
 SAT111690  
 SAT111700  
 SAT111710  
 SAT111720  
 SAT111730  
 SAT111740  
 SAT111750  
 SAT111760  
 SAT111770  
 SAT111780  
 SAT111790  
 SAT111800  
 SAT111810  
 SAT111820  
 SAT111830  
 SAT111840



```

      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.CR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)
      1 IERR=IERR+1
      IF(X(I).LE.XMAX) GC TC 205
      X(I)=XMAX
      GTC 21C
      205 IF(X(I).GE.XMIN) GO TO 210
      X(I)=XMIN
      21C IF(Y(I).LE.YMAX) GO TO 215
      Y(I)=YMAX
      GTC 3C
      215 IF(Y(I).GE.YMIN) GO TO 30
      Y(I)=YMIN
      3C CCNTINUE
      *****
      C** PLCTTING X AND Y AXIS ; IF NECESSARY
      C** *****
      C** *****
      JERR=0
      XRRANGE=XMAX-XMIN
      YRRANGE=YMAX-YMIN
      IF(YRRANGE.NE.0.) GO TO 298
      IF(YMIN.EQ.0.) GO TC 889
      YMIN=0.
      YRRANGE=YMAX
      GC TO 299
      298 IF(XRRANGE.NE.0.) GO TO 295
      IF(XMIN.EQ.0.) GO TC 887
      XMIN=0.
      XRRANGE=XMAX
      *****
      C** BLANKING CUT MATRIX-(GRID)
      C** *****
      C** *****
      255 CC 300 I=1,61
      CC 301 JJ=1,81
      301 GRTC(I,JJ)=BLANK
      CCNTINUE
      IF(XMAX*XMIN.GE.0.) GO TO 222
      IYAXIS=80.*(-XMIN)/XRRANGE+1.5
      CC 40 I=1,61
      40 GRTC(I,IYAXIS)=DOT
      222 IF(YMAX*YMIN.GE.0.) GO TO 333
      IXAXIS=60.*YMAX/YRRANGE+1.5
      CC 60 I=1,81
      6C GRTC(IXAXIS,I)=DOT
      *****
      C** CCMPUTE PROPER SCALE NUMBERS
      C** *****
      C** *****
      323 XINCR=XRRANGE/4.
      YINCR=YRRANGE/6.

```

SAT111850  
 SAT111860  
 SAT111870  
 SAT111880  
 SAT111890  
 SAT111900  
 SAT111910  
 SAT111920  
 SAT111930  
 SAT111940  
 SAT111950  
 SAT111960  
 SAT111970  
 SAT111980  
 \*SAT111990  
 \*SAT112000  
 SAT112010  
 SAT112020  
 SAT112030  
 SAT112040  
 SAT112050  
 SAT112060  
 SAT112070  
 SAT112080  
 SAT112090  
 SAT112100  
 SAT112110  
 SAT112120  
 SAT112130  
 \*SAT112140  
 \*SAT112150  
 SAT112160  
 SAT112170  
 SAT112180  
 SAT112190  
 SAT112200  
 SAT112210  
 SAT112220  
 SAT112230  
 SAT112240  
 SAT112250  
 SAT112260  
 SAT112270  
 SAT112280  
 \*SAT112290  
 \*SAT112300  
 SAT112310  
 SAT112320















```

1 YDEGMS(200),YBSTIF(200),YCSIF(200),YBSTF(20C),YBSTF(20C),
2 YDDSTF(200),YPR(200),YPS(20C),YPT(20C),YPT(20C),YPT(20C),
3 YMT(200),YDTT(20C),YDMT(200),YNS(200),YNTF(20C),
4 YNSTH(200),YQS(20C),YMS(200),YMT(200),YNSTH(20C),
5 YU(200),YV(200),YW(200),YPTF(200),YPTF(200),
6 YPHI(200),XSTATN(200)
DIMENSION PTF(500),PF(500)
*****
1C1 AEZC=SIGO/ELAST
1C2 IF ($DYNMC) GO TO 181
1C3 WRITE(6,101) LSTEP,ALOAD,ITR
1C4 GO TO 182
1C5 DTI=TI*TEEC
1C6 WRITE(6,151) LSTEP,TI,DTI,ITR
1C7 LAM=TKN/CHAR
1C8 ENL=1
1C9 AEZ=SIGC*TKN
1C10 AEZ3=ABZ*TKN*TKN/CHAR
1C11 AEZN=CHAR*SIGO/ELAST
1C12 IF (ITRMAX.EQ.1) ENL=0.
1C13 CC2=1.-NU**2
1C14 CC2I=1./CD2
1C15 CPI=1./SI
1C16 CNI=1./CI
1C17 TCLSQI=.5/CELSQ
1C18 ICHKL=IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)
1C19 ICHK2=IABS(IU)+IABS(IV)+IABS(IPHIS)+IABS(IPHIT)
1C20 IF (NTFMAX.EQ.0) GO TO 991
1C21 DC 21 NTH=1,NTHMAX
1C22 DC 1 MN=1,MNMAX
1C23 I1=1+(MN-1)*KMAX2
1C24 I2=I1+1
1C25 U1(MN)=Z(1,I1)
1C26 U2(MN)=Z(1,I2)
1C27 V1(MN)=Z(2,I1)
1C28 V2(MN)=Z(2,I2)
1C29 W1(MN)=Z(3,I1)
1C30 W2(MN)=Z(3,I2)
1C31 TFFT=TF(NTF)
1C32 WRITE(6,116) THET
1C33 DC 121 K=1,KMAX
1C34 K1=K+1
1C35 CALL BCB(K,BS,DB,CS,DC)
1C36 IF (K.EQ.1.AND.IBCINL.LT.0) CALL PCLE(K,P,DEE,CST,X,Z,20,Z2,Z3,
1C37 ZCCT,IS,JS,ID,JD,PHIXB,PHITE)

```



```

IF(K.EQ.1.AND.IBCINL.LT.O) GO TC 595
IF(K.EQ.KMAX.AND.IBCFNL.LT.O) CALL POLE(K,P,CEE,DST,X,Z,ZO,Z2,Z3,
1 ZCCT,IS,JS,ID,JD,PHIXB,PHITB)
IF(K.EQ.KMAX.AND.IBCFNL.LT.O) GO TO 999
CALL PHIBET(K,Z,IS,JS,ID,JD,PHIXB,PHITB)
CEX=DEGMX(K)
FRA=1./R(K)
CX=CMXI(K)
CT=CT(K)
GA=GAM(K)
DCXT=GX-CT
CCC=GA*COXT
CC2C=CC2*DS
CC 3 MN=1,MNMAXD
EN=N(MN)
ENR=EN*FRA
CALL TLOAD(K,Z)
TTS=TT(MN)*ALOAD
EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))
ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))
EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SCE*BXT3(MN)
1)
KT=ENR*PHIT(MN)+GA*PHIX(MN)
KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GCO*V2(MN)
1 +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-CCXT*PHI(MN))
TX(MN)=ES*(EX+NU*ET)-TTS
TTH(MN)=BS*(DI*EXT
TXI(MN)=BS*(DI*EXT
MKI=KI+(MN-1)*KMAX2
MX(MN)=Z(4,MKI)
MTF(MN)=NU*MX(MN)+DC2D*KT-DI*MT(MN)*ALOAD
MXI(MN)=DS*DI*KXT
MKI1=MKI+1
MKKI=MKI-1
CS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MKI1)-Z(4,MKKI1))*TDLI
1 +ENR*MXI(MN)-GA*MTTH(MN))
MX(MN)=MX(MN)*ABZ3
MTF(MN)=MTTH(MN)*ABZ3
MXI(MN)=MXI(MN)*ABZ3
TX(MN)=TX(MN)*ABZ
TTH(MN)=TTH(MN)*ABZ
TXI(MN)=TXI(MN)*ABZ
PHIX(MN)=PHIX(MN)*ABZO
PHIT(MN)=PHIT(MN)*ABZC
PHI(MN)=PHI(MN)*ABZO
L1(MN)=L2(MN)
L2(MN)=L3(MN)
V1(MN)=V2(MN)

```

SAT114250  
SAT114260  
SAT114270  
SAT114280  
SAT114290  
SAT114300  
SAT114310  
SAT114320  
SAT114330  
SAT114340  
SAT114350  
SAT114360  
SAT114370  
SAT114380  
SAT114390  
SAT114400  
SAT114410  
SAT114420  
SAT114430  
SAT114440  
SAT114450  
SAT114460  
SAT114470  
SAT114480  
SAT114490  
SAT114500  
SAT114510  
SAT114520  
SAT114530  
SAT114540  
SAT114550  
SAT114560  
SAT114570  
SAT114580  
SAT114590  
SAT114600  
SAT114610  
SAT114620  
SAT114630  
SAT114640  
SAT114650  
SAT114660  
SAT114670  
SAT114680  
SAT114690  
SAT114700  
SAT114710  
SAT114720



```

V2(MN)=V3(MN)
W1(MN)=W2(MN)
W2(MN)=W3(MN)
FK=K-1
FIFREQ=IFREQ
FKTST=(K-1)/IFREQ
FKTST=K TST
FKTEST=FK/FIFREQ-FKTST
IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999
IF(FKTEST.NE.0.) GO TO 2
SSS
X(1,K)=C.
X(2,K)=C.
X(3,K)=0.
X(4,K)=C.
PTF(K)=C.
PF(K)=0.
ANX=0.
ANTF=0.
ANTH=0.
ANXTH=0.
ANTF=0.
ANTH=0.
ANXTH=0.
ACS=0.
IF(JUMP.EQ.2) GO TO 73
DC 72 MN=1,MNMAXO
EN=N(MN)
FC=EN*THET
CS=SIN(FC)
CS=cos(FC)
X(1,K)=X(1,K)+U1(MN)*CS*ABZN
X(2,K)=X(2,K)+V1(MN)*SN*ABZN
X(3,K)=X(3,K)+W1(MN)*CS*ABZN
X(4,K)=X(4,K)+PHIX(MN)*CS
PTF(K)=PTF(K)+PHIT(MN)*SN
ANX=ANX+MX(MN)*CS
ANTH=ANTH+MTH(MN)*CS
ANXTH=ANXTH+MXT(MN)*SN
ANX=ANX+IX(MN)*CS
ANTH=ANTH+TTH(MN)*CS
ACS=ACS+QS(MN)*CS
ANXTH=ANXTH+TXT(MN)*SN
PF(K)=PF(K)+PHI(MN)*SN
C***
GC 72 IO 429
CC 74 MN=3,MNMAXO,JUMP
EN=N(MN)
FC=EN*THET
SN=SIN(FC)
SAT114730
SAT114740
SAT114750
SAT114760
SAT114770
SAT114780
SAT114790
SAT114800
SAT114810
SAT114820
SAT114830
SAT114840
SAT114850
SAT114860
SAT114870
SAT114880
SAT114890
SAT114900
SAT114910
SAT114920
SAT114930
SAT114940
SAT114950
SAT114960
SAT114970
SAT114980
SAT114990
SAT115000
SAT115010
SAT115020
SAT115030
SAT115040
SAT115050
SAT115060
SAT115070
SAT115080
SAT115090
SAT115100
SAT115110
SAT115120
SAT115130
SAT115140
SAT115150
SAT115160
SAT115170
SAT115180
SAT115190
SAT115200

```



```

CS=CCS(FC)
MNM=MN-1
X(1,K)=X(1,K)+(U1(MN)*CS+U1(MNM)*SN)*ABZN
X(2,K)=X(2,K)+(V1(MN)*CS+V1(MNM)*CS)*ABZN
X(3,K)=X(3,K)+(W1(MN)*CS+W1(MNM)*SN)*ABZN
X(4,K)=X(4,K)+PHIX(MN)*CS+PHIX(MNM)*SN
PTF(K)=PTF(K)+PHIT(MN)*SN+PHIT(MNM)*CS
AMX=AMX+MX(MN)*CS+MX(MNM)*SN
AMTH=AMTH+MTH(MN)*CS+MTH(MNM)*SN
AMXT=AMXT+MXT(MN)*CS+MXT(MNM)*CS
ANX=ANX+TX(MN)*CS+TX(MNM)*SN
ANTH=ANTH+TTH(MN)*CS+TTH(MNM)*SN
ANXTH=ANXTH+TXT(MN)*SN+TXT(MNM)*CS
ACS=ACS+QS(MN)*CS+QS(MNM)*SN
PF(K)=PF(K)+PHI(MN)*SN+PHI(MNM)*CS
X(1,K)=X(1,K)+U1(1)*ABZN
X(2,K)=X(2,K)+V1(1)*ABZN
X(3,K)=X(3,K)+W1(1)*ABZN
X(4,K)=X(4,K)+PHIX(1)
PTF(K)=PTF(K)+PHIT(1)
PF(K)=PF(K)+PHI(1)
AMX=AMX+MX(1)
AMTH=AMTH+MTH(1)
AMXT=AMXT+MXT(1)
ANX=ANX+TX(1)
ANTH=ANTH+TTH(1)
ANXTH=ANXTH+TXT(1)
ACS=ACS+QS(1)

```

74

C\*\*\*\*\*

```

425 CCNTINUE
IF(K.EQ.1) WRITE(6,117)
WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH
IF ($MOD(CAL.CR.(ICFCK1.EQ.0))) GO TO 2
YNS(K)=ANX
YNTF(K)=ANTH
YNSTH(K)=ANXTH
YCS(K)=AQS
YMS(K)=AMX
YMTF(K)=AMTH
YNSTH(K)=AMXTH
CCNTINUE
121 CC 66C K=1,KMAX
FK=K-1
FIFREQ=IFREQ
KTST=(K-1)/IFREQ
FKTST=KTST
FKTEST=FK/FIFREQ-FKTST

```

SAT115210  
SAT115220  
SAT115230  
SAT115240  
SAT115250  
SAT115260  
SAT115270  
SAT115280  
SAT115290  
SAT115300  
SAT115310  
SAT115320  
SAT115330  
SAT115340  
SAT115350  
SAT115360  
SAT115370  
SAT115380  
SAT115390  
SAT115400  
SAT115410  
SAT115420  
SAT115430  
SAT115440  
SAT115450  
SAT115460  
SAT115470  
SAT115480  
SAT115490  
SAT115500  
SAT115510  
SAT115520  
SAT115530  
SAT115540  
SAT115550  
SAT115560  
SAT115570  
SAT115580  
SAT115590  
SAT115600  
SAT115610  
SAT115620  
SAT115630  
SAT115640  
SAT115650  
SAT115660  
SAT115670  
SAT115680



```

661 IF (K.EC.1.OR.K.EQ.KMAX) GO TO 661
IF (FKTEST.NE.0.) GO TO 658
IF (K.EC.1) WRITE(6,217)
WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),FTF(K),PF(K)
IF ($MOCAL.OR.(ICFCK2.EQ.0)) GO TO 658
YL(K)=X(1,K)
YV(K)=X(2,K)
YK(K)=X(3,K)
YPHIS(K)=X(4,K)
YPHIT(K)=PTF(K)
YFTI(K)=PF(K)
DC 659 I=1,4
X(I,K)=0.
CCNT INUE
IF ($PLCITS.AND..NOT.$MOCAL.AND.((ICCHK1.GT.0).OR.(ICFCK2.GT.0)))
1 CALL PLOT2(NTH)
21 CCNT INUE LE.0) RETURN
CC 534 MN=1,MNMAXQ
WRITE(6,749) N(MN)
DC 521 MN=1,MNMAXC
I1=1+(MN-1)*KMAX2
I2=I1+1
U1(MN)=Z(1,I1)
U2(MN)=Z(1,I2)
V1(MN)=Z(2,I1)
V2(MN)=Z(2,I2)
W1(MN)=Z(3,I1)
W2(MN)=Z(3,I2)
CCNT INUE
DC 445 K=1, KMAX
K1=K+1
CALL BCE(K,BS,DB,CS,DC)
IF (K.EQ.1.AND.IBCINL.LT.0) CALL FCLE(K,P,DEE,CST,X,Z,ZO,Z2,Z3,
1 ZCCT,IS,JS,ID,JD,PHIXB,PHITB)
IF (K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K,P,DEE,DST,X,Z,ZO,Z2,Z3,
1 ZCCT,IS,JS,ID,JD,PHIXB,PHITB)
TXZ=TX(MN)
THZ=TH(MN)
TXTZ=TX(MN)
AMXZ=MX(MN)
AMTHZ=TH(MN)
AMXTZ=MX(MN)
CSZ=QS(MN)
X(1,K)=PHIX(MN)
X(2,K)=PHIT(MN)
X(3,K)=PHI(MN)
IF (K.EQ.1.AND.IBCINL.LT.0) GO TO 583
SAT115690
SAT115700
SAT115710
SAT115720
SAT115730
SAT115740
SAT115750
SAT115760
SAT115770
SAT115780
SAT115790
SAT115800
SAT115810
SAT115820
SAT115830
SAT115840
SAT115850
SAT115860
SAT115870
SAT115880
SAT115890
SAT115900
SAT115910
SAT115920
SAT115930
SAT115940
SAT115950
SAT115960
SAT115970
SAT115980
SAT115990
SAT116000
SAT116010
SAT116020
SAT116030
SAT116040
SAT116050
SAT116060
SAT116070
SAT116080
SAT116090
SAT116100
SAT116110
SAT116120
SAT116130
SAT116140
SAT116150
SAT116160

```



IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GC TC 583  
 CALL PHIBET(K,Z,IS,JS,IO,JD,PHIXB,PHITB)

CEX=DECIM(K)  
 RRA=1./R(K)  
 CX=CMXI(K)  
 CT=CMT(K)  
 GA=GAM(K)  
 CCXT=OX-CT  
 GCC=GA\*CCXT  
 DLZC=CC2\*DS  
 ENR=EN\*RRRA  
 CALL TLCAD(K,Z)  
 TTS=TT(MN)\*ALOAD  
 EX=(U3(MN)-U1(MN))\*TDLI+OX\*W2(MN)+ENL\*OSE\*(EX3(MN)+BE3(MN))  
 ET=ENR\*V2(MN)+GA\*U2(MN)+OT\*W2(MN)+ENL\*OSE\*(ET3(MN)+BE3(MN))  
 EXT=.5\*((V3(MN)-V1(MN))\*TDLI-ENR\*U2(MN)-GA\*V2(MN)+ENL\*SOE\*BX12(MN))  
 1) KT=ENR\*PHIT(MN)+GA\*PHIX(MN)  
 KXT=.5\*(ENR\*(-PHIX(MN)-GA\*W2(MN)+(W3(MN)-W1(MN))\*TDLI)+GCC\*V2(MN)  
 1 +OT\*(V3(MN)-V1(MN))\*TDLI-GA\*PHIT(MN)-CCXT\*PHI(MN))  
 TX2=(BS\*(EX+NU\*ET)-TTS)\*ABZ  
 THZ=(BS\*(ET+NU\*EX)-TTS)\*ABZ  
 TXZ=BS\*D1\*EXT\*ABZ  
 WK1=K1+(MN-1)\*KMAX2  
 AMXZ=Z(4,MK1)  
 AMTHZ=NU\*AMXZ+DD2D\*KI-DI\*MT(MN)\*ALCAD  
 AMXTZ=CS\*CI\*KXT  
 MK11=MK1+1  
 MKK1=MK1-1  
 CSZ=SIGO\*TKN\*LAM2\*(GA\*AMXZ+(Z(4,MK11)-Z(4,MKK1))\*TDLI+ENR\*AMXTZ  
 1 -GA\*AMTHZ)  
 AMXZ=AMXZ\*ABZ3  
 AMTHZ=AMTHZ\*ABZ3  
 AMXTZ=AMXTZ\*ABZ3  
 X(1,K)=PHIX(MN)\*ABZC  
 X(2,K)=PHIT(MN)\*ABZC  
 X(3,K)=PHI(MN)\*ABZC  
 CC533 MM=1,MNMAXC  
 L1(MM)=L2(MM)  
 L2(MM)=U3(MM)  
 L3(MM)=V2(MM)  
 V2(MM)=V3(MM)  
 W1(MM)=W2(MM)  
 W2(MM)=W3(MM)  
 FK=K-1  
 FIFREQ=IFREQ  
 K1ST=(K-1)/IFREQ

533



```

FKTST=KKTST
FKTEST=FK/FIFREQ-FKTST
IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583
IF(FKTEST.NE.0.) GC TC 445
583 CCNTINUE
IF(K.EQ.1) WRITE(6,117)
117 WRITE(6,118) K, TXZ, TTHZ, TXTZ, QSZ, AMXZ, AMT+Z, AMXTZ GC TC 445
118 K, TXZ, TTHZ, TXTZ, QSZ, AMXZ, AMT+Z, AMXTZ
IF(.NOT.$PLOTS.OR..NOT.$MODAL.CR.(ICHECK1.EC.0)) GC TC 445
YAS(K)=TXZ
YATF(K)=TTHZ
YASTH(K)=THTZ
YCS(K)=CSZ
YMS(K)=AMXZ
YMTF(K)=AMTHZ
YMSTH(K)=AMXTZ
CCNTINUE
445 WRITE(6,217)
446 DC 447 K=1, KMAX
FK=K-1
FIFREQ=IFREQ
K1ST=(K-1)/IFREQ
FKTST=K1ST
FKTEST=FK/FIFREQ-FKTST
IF(K.EQ.1.CR.K.EQ.KMAX) GO TO 593
IF(FKTEST.NE.0.) GO TO 447
593 KZ=K+1+(MN-1)*KMAX2
LF=Z(1,KZ)*ABZN
VF=Z(2,KZ)*ABZN
WP=Z(3,KZ)*ABZN
WRITE(6,218) K, UP, VP, WP, X(1,K), X(2,K), X(3,K)
IF(.NOT.$PLOTS.OR..NOT.$MODAL.CR.(ICHECK2.EC.0)) GC TC 447
YL(K)=UP
YV(K)=VP
YW(K)=WF
YFIS(K)=X(1,K)
YFIT(K)=X(2,K)
YFI(K)=X(3,K)
447 CCNTINUE
IF($PLOTS.AND.$MODAL.AND.((ICHECK1.GT.0).OR.(ICHECK2.GT.0)))
1 CALL PLCT2(1)
534 CCNTINUE
C*****
101 FCRMAT(1,1) THE LCAC STEP NUMBER IS '12,1'
102 FCRMAT(1,2) THE SOLUTION CONVERGED IN '12,1'
103 FCRMAT(1,3) THE SUMMED FCRCES, MCMENTS, DISPLACEMENTS AN
104 FCRMAT(1,4) FOLLOW FOR THETA ='E15.6///' N S
105 FCRMAT(1,5) STATION N THETA
106 FCRMAT(1,6) N ST-ETA
117 FCRMAT(1,7) N ST-ETA

```











```

1  WZ(M)=Z(3,11)
   IF(IBCINL.LT.0) GC TO 100
   CALL PHIBET(1,Z,IS,JS,ID,JD,PHIXE,PHITB)
   CC 2  N=1,MNMAX
   B>1(M)=BX3(M)
   B>11(M)=BT3(M)
   B>111(M)=BX73(M)
   BE1(M)=BE3(M)
   CC 3  TEAETA(1,Z,IS,JS,ID,JD)
   CC 3  M=1,MNMAX
   EXX1(M)=EXX3(M)
   ETT1(M)=ETT3(M)
   ETX1(M)=ETX3(M)
   EXT1(M)=EXT3(M)
   ETT1(M)=ETT3(M)
   CC 4  PHIBET(2,Z,IS,JS,ID,JD,PHIXE,PHITB)
   CC 4  N=1,MNMAX
   BX2(M)=BX3(M)
   ET2(M)=BT3(M)
   EXT2(M)=BX73(M)
   BE2(M)=BE3(M)
   CC 5  TEAETA(2,Z,IS,JS,ID,JD)
   CC 5  N=1,MNMAX
   EXX2(M)=EXX3(M)
   ETT2(M)=ETT3(M)
   ETX2(M)=ETX3(M)
   EXT2(M)=EXT3(M)
   ET2(M)=ETT3(M)
   CC 6  PHIBET(3,Z,IS,JS,ID,JD,PHIXE,PHITB)
   CC 6  TEAETA(3,Z,IS,JS,IC,JD)
   CCNT INUE
   IF(IBCINL.LT.0) GC TO 20
   CALL BOB(1,B1,DB,D,CD)
   GAM1=GAM(1)
   CC 8  TLCCAD(1,Z)
   CC 8  N=1,MNMAX
   IF(ITRMAX.EQ.1) GO TO 67
   FFS(1,M)=-TT(M)*ALCAD+OSE*(BX1(M)+BE1(M)+NU*(BT1(M)+BE1(M))*B1
   FFS(2,M)=OSE*(B1*CI*BT1(M)+EX1(M)+ETT1(M))
   FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD-(EXX1(M)+ETX1(M))*SOE
   GC TO 8
   CC 9  FFS(1,M)=-TT(M)*ALCAD
   FFS(2,M)=0
   FFS(3,M)=LAM2*GAM1*CI*MT(M)*ALOAD
   FFS(4,M)=0
   CC 9  I=1,4

```

```

SAT118090
SAT118100
SAT118110
SAT118120
SAT118130
SAT118140
SAT118150
SAT118160
SAT118170
SAT118180
SAT118190
SAT118200
SAT118210
SAT118220
SAT118230
SAT118240
SAT118250
SAT118260
SAT118270
SAT118280
SAT118290
SAT118300
SAT118310
SAT118320
SAT118330
SAT118340
SAT118350
SAT118360
SAT118370
SAT118380
SAT118390
SAT118400
SAT118410
SAT118420
SAT118430
SAT118440
SAT118450
SAT118460
SAT118470
SAT118480
SAT118490
SAT118500
SAT118510
SAT118520
SAT118530
SAT118540
SAT118550
SAT118560

```



```

5  ELIS(I)=ALOAD*ELI(I)
20 CALL FORCE(1,P,X,DEE,DST,Z,ZO,Z2,Z3)
   CALL FCRCE(2,P,X,DEE,DST,Z,ZO,Z2,Z3)
   CC 10 K=3,KLL
   KF=K+1
   IF(ITRMAX.EQ.1) GO TO 1C
   CALL UPDATE
   CALL PHIBET(KP,Z,IS,JS,ID,JD,PHIXE,PHITB)
   CALL TEAETA(KP,Z,IS,JS,ID,JD)
   CALL FCRCE(K,P,X,DEE,DST,Z,ZO,Z2,Z3)
1C  IF(ITRMAX.NE.1) CALL UPDATE
   IF(IBCFL.LT.0) GO TO 120
   IF(ITRMAX.EQ.1) GO TO 11
   CALL PHIBET(KMAX,Z,IS,JS,ID,JD,PHIXB,PHITB)
   CALL TEAETA(KMAX,Z,IS,JS,ID,JD)
11  CALL FCRCE(KL,P,X,DEE,DST,Z,ZO,Z2,Z3)
   CALL FORCE(KMAX,P,X,DEE,DST,Z,ZO,Z2,Z3)
12  CC 12 I=1,4
   ELLS(I)=ALCAD*ELL(I)
   CALL BCE(KMAX,BL,DB,O,DD)
   GAML=GAM(KMAX)
   FLS(4)=C
   CALL TLCAD(KMAX,Z)
   CC 14 N=1,MNMAX
   IF(M.GT.1) ELLS(1)=0.0
   IF(ITRMAX.EQ.1) GC TO 68
   FLS(1)=-TT(M)*ALCAC+QSE*(BX3(M)+BE3(M)+NU*(ET3(M)+BE3(M)))*BL
   FLS(2) =QSE*(BL*DI*8XT3(M)+EX3(M)+ET3(M))
   FLS(3) =LAM2*GAML*DI*MT(M)*ALOAD-(EXX3(M)+ETX3(M))*SOE
   GC TO 65
68  FLS(1)=-TT(M)*ALOAD
   FLS(2)=C
   FLS(3)=LAM2*GAML*DI*MT(M)*ALOAD
69  CC CONTINUE
   IK=KL+KMAX*(M-1)
   IJ=KMAX*M
   L=M*KMAX2
   CC 14 I=1,4
   SUMZ=0.
   CC 15 J=1,4
   C*****
   C THE FOLLOWING CARD CAUSES BCUNDARY CONS TO EXIST FCR MODE 'O' ONLY
   C*****
   IF (M.NE.1) ELLS(J)=0
15  SUMZ=SUMZ+ZF1M(I,J,M)*ELLS(J)+ZF2M(I,J,M)*X(J,IJ)+ZF3M(I,J,M)*
14  X(J,IK)+ZF4M(I,J,M)*FLS(J)
   Z(I,L)=SUMZ
   Z(I)=1

```

SAT18570  
 SAT18600  
 SAT18610  
 SAT18620  
 SAT18630  
 SAT18640  
 SAT18650  
 SAT18670  
 SAT18680  
 SAT18690  
 SAT18700  
 SAT18710  
 SAT18740  
 SAT18750  
 SAT18760  
 SAT18770  
 SAT18780  
 SAT18790  
 SAT18800  
 SAT18810  
 SAT18820  
 SAT18830  
 SAT18840  
 SAT18850  
 SAT18860  
 SAT18870  
 SAT18880  
 SAT18890  
 SAT18900  
 SAT18910  
 SAT18920  
 SAT18930  
 SAT18940  
 SAT18950  
 SAT18960  
 SAT18970  
 SAT18980  
 SAT18990  
 SAT19000  
 SAT19010  
 SAT19020  
 SAT19030  
 SAT19040



```

150 CC 16 M=1, MNMAX
CC 16 L=L3, KMAX
K=KMAX2-L
KFX=K-1
KZ=K+1
JJ=KPX+(M-1)*KMAX
JK=KZ+(M-1)*KMAX2
KK=JK-1
CC 17 I=1, 4
SUMZ=0.
CC 18 J=1, 4
SUMZ=SUMZ-P(I,J,IJ)*Z(J,JK)
SUMZ=SUMZ+X(I,I,IJ)
ASUMZ=ABS(SUMZ)
IF(ASUMZ.GT.1.E+15) ITR=ITRMAX
IF(ASUMZ.NE.1. OR. ASUMZ.LT. 1.E-05) GO TO 17
DELZ=ABS(Z(I,KK)-SUMZ)
ZTEST=EPS*TZMAX(I,M)
IF(DELZ.GT.ZTEST) NCONV=0
Z(I,KK)=SUMZ
17 CCATINCE
16 IF(IBCINL.LT.0) GO TO 30
CC 25 M=1, MNMAX
CALL EFG(I,M,ZO,Z2,Z3)
CALL ABC
IJ=2+(M-1)*KMAX2
I-1=IJ+1
I-2=IJ-1
CC 21 I=1, 4
SUMZ=0.
CC 22 J=1, 4
SUMZ=SUMZ-A(I,J)*Z(J,IJ1)-BEE(I,J)*Z(J,IJ)
22 ZT(I)=SUMZ+GEES(I,M)
21 CALL MATINV(C,4,Z1,1, DETERM, IPIVOT, INDEX, 4, ISCALE)
23 CC 23 I=1, 4
Z(I,IJ2)=ZT(I)
23 CCATINCE
23 RETURN
100 CALL INLPOL (Z, PHIXB, PHITB)
CC 101 M=1, MNMAXC
L1(M)=U2(M)
V1(M)=V2(M)
W1(M)=W2(M)
IJ=3+KMAX2*(M-1)
L2(M)=Z(1,IJ)
V2(M)=Z(2,IJ)
W2(M)=Z(3,IJ)
101 GC TO 102

```

SAT119050  
SAT119060  
SAT119070  
SAT119080  
SAT119090  
SAT119100  
SAT119110  
SAT119120  
SAT119130  
SAT119140  
SAT119150  
SAT119160  
SAT119170  
SAT119180  
SAT119190  
SAT119200  
SAT119210  
SAT119220  
SAT119230  
SAT119240  
SAT119250  
SAT119260  
SAT119270  
SAT119280  
SAT119290

SAT119310  
SAT119320  
SAT119330  
SAT119340  
SAT119350  
SAT119360  
SAT119370  
SAT119380  
SAT119390  
SAT119400  
SAT119410  
SAT119420  
SAT119430  
SAT119440  
SAT119450  
SAT119460  
SAT119470  
SAT119480  
SAT119490  
SAT119500  
SAT119510  
SAT119520



```

120 IF (ITRMX.NE.1) CALL FNLPOL (Z,P,IXB,PHITB)
    CALL FCRCE(KL,P,X,DEE,DST,Z,ZC,Z2,Z3)
    IF (M2.EC.0) GO TO 122
    L=KL+(M2-1)*KMAX
    LI=KMAX1+(M2-1)*KMAX2
    CC 130 I=1,4
    SUM=0.
    CC 121 J=1,4
    SUM=SUM+CL2(I,J)*X(J,L)
    ASUMZ=ABS(SUM)
    IF (NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TC 130
    CELZ=ABS(Z(I,LI)-SUM)
    ZTEST=EPS*TZMAX(I,M2)
    IF (CELZ.GT.ZTEST) NCONV=0
    12C Z(I,LI)=SUM
    122 IF (M1.EC.0) GO TO 123
    L=KL+(M1-1)*KMAX
    LI=KMAX1+(M1-1)*KMAX2
    CC 132 I=1,4
    SUM=0.
    CC 133 J=1,4
    SUM=SUM+CL1(I,J)*X(J,L)
    ASUMZ=ABS(SUM)
    IF (NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TC 132
    CELZ=ABS(Z(I,LI)-SUM)
    ZTEST=EPS*TZMAX(I,M1)
    IF (CELZ.GT.ZTEST) NCONV=0
    132 Z(I,LI)=SUM
    123 IF (M0.EC.0) GO TO 124
    L=KL+(M0-1)*KMAX
    LI=KMAX1+(M0-1)*KMAX2
    CC 134 I=1,4
    SUM=0.
    CC 135 J=1,4
    SUM=SUM+CLC(I,J)*X(J,L)
    ASUMZ=ABS(SUM)
    IF (NCONV.NE.1 .OR. ASUMZ.LT.1.E-06) GO TC 134
    CELZ=ABS(Z(I,LI)-SUM)
    ZTEST=EPS*TZMAX(I,M0)
    IF (CELZ.GT.ZTEST) NCONV=0
    134 Z(I,LI)=SUM
    124 LC=2
    GC TO 150
    ENC

SLROUTINE PLOT2(NTN)
C*****
C THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER *****
SAT19530
SAT19550
SAT19560
SAT19570
SAT19580
SAT19590
SAT19600
SAT19610
SAT19620
SAT19630
SAT19640
SAT19650
SAT19660
SAT19670
SAT19680
SAT19690
SAT19700
SAT19710
SAT19720
SAT19730
SAT19740
SAT19750
SAT19760
SAT19770
SAT19780
SAT19790
SAT19800
SAT19810
SAT19820
SAT19830
SAT19840
SAT19850
SAT19860
SAT19870
SAT19880
SAT19890
SAT19900
SAT19910
SAT19920
SAT19930
SAT19940
SAT19950
SAT19960
SAT19970
SAT19980
SAT19990
*****SAT20000

```







```

      WRITE (6,1006) TH(NTH)
      IF (IMSTH.EQ.0) GO TO 1211
      WRITE (6,1000)
      IF (IMSTH.GT.0) CALL PLCTIT (XSTATN,YMSTH,KMAX,0)
      IF (IMSTH.LT.0) CALL PLCTIT (XSTATN,YMSTH,NGKMAX,0)
      WRITE (6,1007) TH(NTH)
      IF (IU.EQ.0) GO TO 10
      IF (IU.EQ.0) GO TO 10
      WRITE (6,1000)
      IF (IU.GT.0) CALL PLCTIT (XSTATN,YU,KMAX,0)
      IF (IU.LT.0) CALL PLCTIT (XSTATN,YU,NGKMAX,C)
      WRITE (6,1010) TF(NTH)
      IF (IV.EQ.0) GO TO 11
      IF (IV.EQ.0) GO TO 11
      WRITE (6,1000)
      IF (IV.GT.0) CALL PLCTIT (XSTATN,YV,KMAX,C)
      IF (IV.LT.0) CALL PLCTIT (XSTATN,YV,NGKMAX,C)
      WRITE (6,1009) TH(NTH)
      IF (IW.EQ.0) GO TO 12
      IF (IW.EQ.0) GO TO 12
      WRITE (6,1000)
      IF (IW.GT.0) CALL PLCTIT (XSTATN,YW,KMAX,0)
      IF (IW.LT.0) CALL PLCTIT (XSTATN,YW,NGKMAX,0)
      WRITE (6,1008) TF(NTH)
      IF (IPHS.EQ.0) GO TO 13
      IF (IPHS.EQ.0) GO TO 13
      WRITE (6,1000)
      IF (IPHS.GT.0) CALL PLCTIT (XSTATN,YPHIS,KMAX,0)
      IF (IPHS.LT.0) CALL PLCTIT (XSTATN,YPHIS,NGKMAX,0)
      WRITE (6,1011) TH(NTH)
      IF (IPHT.EQ.0) GO TO 14
      IF (IPHT.EQ.0) GO TO 14
      WRITE (6,1000)
      IF (IPHT.GT.0) CALL PLCTIT (XSTATN,YPHIT,KMAX,0)
      IF (IPHT.LT.0) CALL PLCTIT (XSTATN,YPHIT,NGKMAX,0)
      WRITE (6,1012) TH(NTH)
      IF (IPFI.EQ.0) GO TO 21
      IF (IPFI.EQ.0) GO TO 21
      WRITE (6,1000)
      IF (IPHI.GT.0) CALL PLCTIT (XSTATN,YPHI ,KMAX,0)
      IF (IPHI.LT.0) CALL PLCTIT (XSTATN,YPHI ,NGKMAX,0)
      WRITE (6,1013) TF(NTH)
      RETURN
21 IF (INS.EQ.0) GO TO 15
121 IF (INS.EQ.0) GO TO 15
      WRITE (6,1000)
      IF (INS.GT.0) CALL PLCTIT (XSTATN,YNS,KMAX,C)
      IF (INS.LT.0) CALL PLCTIT (XSTATN,YNS,NGKMAX,0)
      WRITE (6,2001)
      IF (INTF.EQ.0) GO TO 16
      IF (INTF.EQ.0) GO TO 16
      WRITE (6,1000)
      IF (INTF.GT.0) CALL PLCTIT (XSTATN,YNTH,KMAX,C)
      IF (INTF.LT.0) CALL PLCTIT (XSTATN,YNTH,NGKMAX,0)
      WRITE (6,2002)
      IF (INSTH.EQ.0) GO TO 17
      IF (INSTH.EQ.0) GO TO 17

```



```

WRITE (6,1000)
IF (INSTH.GT.0) CALL PLCTIIT (XSTATN,YNSTH,KMAX,0)
IF (INSTH.LT.0) CALL PLCTIIT (XSTATN,YNSTH,NGKMAX,0)
17 IF (ICS.EQ.0) GO TO 18
WRITE (6,1000)
IF (IQS.GT.0) CALL PLCTIIT (XSTATN,YQS,KMAX,C)
IF (IQS.LT.0) CALL PLCTIIT (XSTATN,YQS,NGKMAX,C)
18 IF (IMS.EQ.0) GO TO 19
WRITE (6,1000)
IF (IMS.GT.0) CALL PLOTIIT (XSTATN,YMS,KMAX,C)
IF (IMS.LT.0) CALL PLOTIIT (XSTATN,YMS,NGKMAX,0)
19 IF (IMTH.EQ.0) GO TO 22
WRITE (6,1000)
IF (IMTH.GT.0) CALL FLCTIIT (XSTATN,YMTH,KMAX,C)
IF (IMTH.LT.0) CALL FLCTIIT (XSTATN,YMTH,NGKMAX,0)
22 IF (IMSTH.EQ.0) GO TO 231
WRITE (6,1000)
IF (IMSTH.GT.0) CALL PLCTIIT (XSTATN,YMSTH,KMAX,0)
IF (IMSTH.LT.0) CALL PLCTIIT (XSTATN,YMSTH,NGKMAX,0)
231 IF (IU.EQ.0) GO TO 23
WRITE (6,1000)
IF (IU.GT.0) CALL PLOTIIT (XSTATN,YU,KMAX,C)
IF (IU.LT.0) CALL PLOTIIT (XSTATN,YU,NGKMAX,0)
23 IF (IV.EQ.0) GO TO 24
WRITE (6,1000)
IF (IV.GT.0) CALL PLOTIIT (XSTATN,YV,KMAX,C)
IF (IV.LT.0) CALL PLOTIIT (XSTATN,YV,NGKMAX,C)
24 IF (IW.EQ.0) GO TO 25
WRITE (6,1000)
IF (IW.GT.0) CALL PLOTIIT (XSTATN,YW,KMAX,0)
IF (IW.LT.0) CALL PLOTIIT (XSTATN,YW,NGKMAX,C)
25 IF (IPHS.EQ.0) GO TO 26
WRITE (6,1000)
IF (IPHS.GT.0) CALL PLCTIIT (XSTATN,YPHIS,KMAX,0)
IF (IPHS.LT.0) CALL PLCTIIT (XSTATN,YPHIS,NGKMAX,0)
26 IF (IPHIT.EQ.0) GO TO 27
WRITE (6,1000)
IF (IPHIT.GT.0) CALL PLCTIIT (XSTATN,YPHIT,KMAX,0)
IF (IPHIT.LT.0) CALL PLCTIIT (XSTATN,YPHIT,NGKMAX,0)

```

SAT2097C  
 SAT20980  
 SAT20990  
 SAT21000  
 SAT21010  
 SAT21020  
 SAT21030  
 SAT2104C  
 SAT21050  
 SAT21060  
 SAT21070  
 SAT21080  
 SAT21090  
 SAT21100  
 SAT21110  
 SAT21120  
 SAT21130  
 SAT21140  
 SAT21150  
 SAT21160  
 SAT21170  
 SAT21180  
 SAT21190  
 SAT21200  
 SAT21210  
 SAT21220  
 SAT21230  
 SAT21240  
 SAT21250  
 SAT21260  
 SAT21270  
 SAT21280  
 SAT21290  
 SAT21300  
 SAT21310  
 SAT21320  
 SAT21330  
 SAT21340  
 SAT21350  
 SAT21360  
 SAT21370  
 SAT21380  
 SAT21390  
 SAT21400  
 SAT21410  
 SAT21420  
 SAT21430  
 SAT21440



```

27 WRITE (6,2C12) GO TO 28
   IF (IPHI.EQ.0)
28 WRITE (6,1000)
   IF (IPHI.GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)
   IF (IPHI.LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)
   RETURN
C*****
1000 FCRMAT (,1),T10,"S"-MEMBRANE FCRCE VS STATN, MERIDIAN AT THETA =
1001 FCRMAT (,0, F10.5, RADIANS,)
1002 FCRMAT (,0, T10,"THETA"-MEMBRANE FCRCE VS STATN, MERIDIAN AT THETA =
1003 FCRMAT (,0, F10.5, RADIANS,)
1004 FCRMAT (,0, T10,"S"-THETA" MEMBRANE FORCE VS STATN, MERIDIAN AT THETA =
1005 FCRMAT (,0, F10.5, RADIANS,)
1006 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATN, MERIDIAN AT THETA =
1007 FCRMAT (,0, F10.5, RADIANS,)
1008 FCRMAT (,0, T10,"THETA"-BENDING MOMENT VS STATN, MERIDIAN AT THETA =
1009 FCRMAT (,0, F10.5, RADIANS,)
1010 FCRMAT (,0, T10,"S"-THETA" BENDING MOMENT VS STATN, MERIDIAN AT THETA =
1011 FCRMAT (,0, F10.5, RADIANS,)
1012 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATN, MERIDIAN AT THETA =
1013 FCRMAT (,0, F10.5, RADIANS,)
2001 FCRMAT (,0, T10,"S"-MEMBRANE FCRCE VS STATIGN,)
2002 FCRMAT (,0, T10,"THETA"-MEMBRANE FCRCE VS STATIGN,)
2003 FCRMAT (,0, T10,"S"-THETA" MEMBRANE FORCE VS STATIGN,)
2004 FCRMAT (,0, T10,"S"-THETA" MEMBRANE FORCE VS STATIGN,)
2005 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2006 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2007 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2008 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2009 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2010 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2011 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2012 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)
2013 FCRMAT (,0, T10,"S"-BENDING MOMENT VS STATIGN,)

```



```

C *** ENCLINE PLOT1(I)
C *** S ***** CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER
C *** THIS SUBROUTINE INPUT QUANTITIES
C *** SPECIFIED) *****
C *** INFLICIT LOGICAL*1 ($)
C *** CCMCN /IBL4/ KMAX, KL
C *** CCMCN /BLPLOT/
C *** 1 IRADII, IGAMMA, ICMEGS, IOMEGT, IDEOMS, IBSTIF, IDSTIF,
C *** 2 IBBSTF, IDDSIF, IPR, IPS, IPT, IIT, IMT, IDIT, IDMT, INS,
C *** 3 INTH, INTH, IQS, IMS, IMTH, IMSTH, IU, IV, IW, IFFIS,
C *** 4 IPHI, IPHI, $PLOTS, $MODAL
C *** 5 XRADII(200), YGAMMA(200), YCMEGS(200), YCMEGT(200),
C *** 6 YDEQMS(200), YBSTIF(200), YDSTIF(200), YBBSTIF(200),
C *** 1 YDCSTF(200), YPR(200), YPS(200), YPT(200), YTT(200),
C *** 2 YMT(200), YDIT(200), YDMT(200), YNS(200), YNTF(200),
C *** 3 YNSTH(200), YQS(200), YMS(200), YMTH(200), YNSTH(200),
C *** 4 YU(200), YV(200), YW(200), YPHIS(200), YPHIT(200),
C *** 5 YPHI(200), XSTATN(200)
C *** NGKMAX=-KMAX
C *** IF (I.GT.1) GO TO 1 GC TO 1
C *** IF (IRADII.EQ.0) GC TO 2
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,XRADII,NGKMAX,0)
C *** WRITE (6,1001)
C *** IF (IGAMMA.EQ.0) GC TO 4
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,YGAMMA,NGKMAX,0)
C *** WRITE (6,1002)
C *** IF (ICMEGS.EQ.0) GC TO 5
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,YOMEGS,NGKMAX,0)
C *** IF (ICMEGT.EQ.0) GC TO 6
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,YCMEGT,NGKMAX,0)
C *** IF (IDECMS.EQ.0) GC TO 7
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,YDEQMS,NGKMAX,0)
C *** IF (IBSTIF.EQ.0) GC TO 8
C *** WRITE (6,1000)
C *** CALL PLCTIT (XSTATN,YBSTIF,NGKMAX,0)
C *** IF (IDSTIF.EQ.0) GC TO 9

```



```

WRITE (6,1000)
CALL PLCTIT (XSTATN,YDSTIF,NGKMAX,0)
WRITE (6,1007)
5 IF (IBBSTF.EQ.0) GO TO 10
WRITE (6,1000)
CALL PLCTIT (XSTATN,YBBSTF,NGKMAX,0)
WRITE (6,1008)
10 IF (IDCSTF.EQ.0) GO TO 1
WRITE (6,1000)
CALL PLCTIT (XSTATN,YDCSTF,NGKMAX,0)
WRITE (6,1009)
1 IF (IPR.EQ.0) GO TO 11
WRITE (6,1000)
CALL PLCTIT (XSTATN,YPR,NGKMAX,0)
WRITE (6,1010)
11 IF (IPS.EQ.0) GO TO 12
WRITE (6,1000)
CALL PLCTIT (XSTATN,YPS,NGKMAX,0)
WRITE (6,1011)
12 IF (IPT.EQ.0) GO TO 13
WRITE (6,1000)
CALL PLCTIT (XSTATN,YPT,NGKMAX,0)
WRITE (6,1012)
13 IF (ITT.EQ.0) GO TO 14
WRITE (6,1000)
CALL PLCTIT (XSTATN,YTT,NGKMAX,0)
WRITE (6,1013)
14 IF (IMT.EQ.0) GO TO 15
WRITE (6,1000)
CALL PLCTIT (XSTATN,YMT,NGKMAX,0)
WRITE (6,1014)
15 IF (IDTT.EQ.0) GO TO 16
WRITE (6,1000)
CALL PLCTIT (XSTATN,YDTT,NGKMAX,0)
WRITE (6,1015)
16 IF (ICMT.EQ.0) GO TO 17
WRITE (6,1000)
CALL PLCTIT (XSTATN,YDMT,NGKMAX,0)
WRITE (6,1016)
17 RETURN
*****
1000 FCFRMT ('1')
1001 FCFRMT ('0',T10,'RADIUS VS STATION')
1002 FCFRMT ('0',T10,'GAMMA VS STATION')
1003 FCFRMT ('0',T10,'OMEGA-S VS STATION')
1004 FCFRMT ('0',T10,'OMEGA-THETA VS STATION')
1005 FCFRMT ('0',T10,'DECMEGA-S VS STATION')

```

SAT222410  
 SAT222420  
 SAT222430  
 SAT222440  
 SAT222450  
 SAT222460  
 SAT222470  
 SAT222480  
 SAT222490  
 SAT222500  
 SAT222510  
 SAT222520  
 SAT222530  
 SAT222540  
 SAT222550  
 SAT222560  
 SAT222570  
 SAT222580  
 SAT222590  
 SAT222600  
 SAT222610  
 SAT222620  
 SAT222630  
 SAT222640  
 SAT222650  
 SAT222660  
 SAT222670  
 SAT222680  
 SAT222690  
 SAT222700  
 SAT222710  
 SAT222720  
 SAT222730  
 SAT222740  
 SAT222750  
 SAT222760  
 SAT222770  
 SAT222780  
 SAT222790  
 SAT222800  
 SAT222810  
 SAT222820  
 SAT222830  
 SAT222840  
 SAT222850  
 SAT222860  
 SAT222870  
 SAT222880



```

10C6 FCRMAT ((0, T10, 'B-STIFFNESS VS STATION')) VS STATION')
10C7 FCRMAT ((0, T10, 'D-STIFFNESS VS STATION')) VS STATION')
10C8 FCRMAT ((0, T10, 'CB-STIFFNESS VS STATION')) VS STATION')
10C9 FCRMAT ((0, T10, 'DC-STIFFNESS VS STATION')) VS STATION')
1010 FCRMAT ((0, T10, 'NORCMAL LOADING VS STATION')) VS STATION')
1011 FCRMAT ((0, T10, 'MERICUMAL LOADING VS STATION')) VS STATION')
1012 FCRMAT ((0, T10, 'CIRCUMAL LOADING VS STATION')) VS STATION')
1013 FCRMAT ((0, T10, 'THERMAL BENDING VS STATION')) VS STATION')
1014 FCRMAT ((0, T10, 'THERMAL BENDING VS STATION')) VS STATION')
1015 FCRMAT ((0, T10, 'DE-THERMAL BENDING VS STATION')) VS STATION')
1016 FCRMAT ((0, T10, 'DE-THERMAL BENDING VS STATION')) VS STATION')
C** ENCLROUTINE FLYNVY
S** WRITE ((6,11))
R** WRITE ((6,1))
R** WRITE ((6,2))
R** WRITE ((6,3))
R** WRITE ((6,4))
R** WRITE ((6,5))
R** WRITE ((6,6))
R** WRITE ((6,7))
R** WRITE ((6,8))
R** WRITE ((6,9))
R** WRITE ((6,10))
R** WRITE ((6,11))
R** WRITE ((6,12))
R** WRITE ((6,13))
R** WRITE ((6,14))
R** WRITE ((6,15))
R** WRITE ((6,16))
R** WRITE ((6,17))
R** WRITE ((6,18))
R** WRITE ((6,19))
R** WRITE ((6,20))
R** WRITE ((6,21))
R** WRITE ((6,22))
R** WRITE ((6,23))
R** WRITE ((6,24))
R** WRITE ((6,25))
R** WRITE ((6,26))
R** WRITE ((6,27))
R** WRITE ((6,28))
R** WRITE ((6,29))
R** WRITE ((6,30))
R** WRITE ((6,31))
R** WRITE ((6,32))
R** WRITE ((6,33))

```







```

20 FCRMAT ( , , T11, , 000 )
XOO X X X COX
21 FCRMAT ( , , T11, , 0000 )
XOC X OCX
22 FCRMAT ( , , T11, , 0000 )
XOCX
23 FCRMAT ( , , T11, , 00000 )
CCC COCXCOX
24 FCRMAT ( , , T11, , 000000 )
CCCCCCCCCOX
25 FCRMAT ( , , T11, , 0000000 )
CCCCCCCCCOX
26 FCRMAT ( , , T11, , 00000000 )
CCCCCCCCCOX
27 FCRMAT ( , , T11, , 000000000 )
CCCCCCCCCOX
28 FCRMAT ( , , T11, , 0000000000 )
CCCCCCCCCOX
29 FCRMAT ( , , T11, , 00000000000 )
CCCCCCCCCOX
30 FCRMAT ( , , T11, , 000000000000 )
CCCCCCCCCOX
31 FCRMAT ( , , T11, , 0000000000000 )
CCCCCCCCCOX
32 FCRMAT ( , , T11, , 00000000000000 )
CCCCCCCCCOX
33 FCRMAT ( , , T11, , 000000000000000 )
CCCCCCCCCOX
34 FCRMAT ( , , T11, , 0000000000000000 )
CCCCCCCCCOX
35 FCRMAT ( , , T11, , 00000000000000000 )
CCCCCCCCCOX
36 FCRMAT ( , , T11, , 000000000000000000 )
CCCCCCCCCOX
37 FCRMAT ( , , T11, , 0000000000000000000 )
CCCCCCCCCOX
38 FCRMAT ( , , T11, , 00000000000000000000 )
CCCCCCCCCOX
39 FCRMAT ( , , T11, , 000000000000000000000 )
CCCCCCCCCOX
40 FCRMAT ( , , T11, , 0000000000000000000000 )
CCCCCCCCCOX
1 RETURN
END
SLEROUTINE PMATRIX ( P, X, Z0, Z2, Z3, CEE, DST )
*****
***** THIS SUBROUTINE CALLS THE SUBROUTINES FJ(K,MN), EFG(K,MN), ABC, 00001310
*****
C

```



```

AND PANDD(K,MN) TO SET UP THE P, P-BAR AND P-HAT MATRICES GIVEN
BY EQUATIONS (30).
INTERNALLY, MATRICES DL, DG AND DF ARE SET UP FOR THE CALCULA-
TION OF X(1) GIVEN BY EQUATION (31A), WHERE
X(1) = DL*SMALL-L(1) + DG*SMALL-G(1) + DF*SMALL-F(1)
THE SPECIAL P MATRIX FOR A SHELL WITH AN INITIAL PCLE IS ALSO
COMPLETED HERE.
MATRICES ZF1M, ZF2M, ZF3M, ZF4M ARE SET UP FOR THE CALCULATION OF
Z(K+1) GIVEN BY EQUATION (31B), WHERE
Z(K+1)=ZF1M*SMALL-L(K) + ZF2M*X(K) + ZF3M*X(K-1) + ZF4M*SMALL-F(1)
IF THE SHELL HAS A FINAL POLE, THE MATRICES CLC, CL1, CL2 ARE
PREPARED FOR THE CALCULATION OF Z(K)
*****
REAL JAY
DIMENSION P(4,4,1), CEE(4,4,1), DST(4,4,1), X(4,1), ZC(4,1),
1 Z2(4,1), Z3(4,1)
CCNMCON /IBL1/ MNMAX
CCNMCON /IBL2/ N(99), MNINIT
CCNMCON /IBL3/ MO,M1,M2,M3
CCNMCON /IBL4/ KMAX,KL
CCNMCON /IBL5/ IBCFNL, IBCFNL
CCNMCON /IBL1/ A(4,4), BEE(4,4), C(4,4)
CCNMCON /IBL4/ ZF1M(4,4,99), ZF2M(4,4,99),
1 ZF3M(4,4,99), ZF4M(4,4,99)
CCNMCON /IBL13/ OMEGL(4,4), CAPL1(4,4), OMEGL(4,4), CAPL1(4,4),
1 OMEGL(4,4), CAPL1(4,4), CAPL1(4,4), CAPL1(4,4), CAPL1(4,4),
1 JAY(4,4), F(4,4), DF(4,4,99)
CCNMCON /BL23/ JAY(4,4), F(4,4), DF(4,4,99)
CCNMCON /BL24/ DL(4,4,99), DG(4,4,99)
CCNMCON /BL25/ E(4,4), F(4,4), G(4,4)
1 CIPENSICN PATA(4,4), PBTA(4,4), POTA(4,4), PJTA(4,4), PTR(4,
2 T(4), CGG(4,4), ZF1(4,4), ZF2(4,4), ZFPC(4,4), ZFP1(4,4), ZFP2(4,4),
2 ECLIVALENCE (CLO(1), ZF1M(1), CL1(4,4), CL2(4,4), CL1(1), ZF2M(1)),
1 (ZFPO(1), PATA(1)), (ZFPL(1), PBTA(1)), (ZFP2(1), POTA(1)),
2 (ZF1(1), DLL(1)), (ZF2(1), PTR(1))
*****
C *****
IF (IBCFNL.LT.0) GO TO 10
CC 1 MN=MNINIT, MNMAX
CALL FJ(1,MN)
CALL EFG(1,MN,ZO,Z2,Z3)
CALL ABC
CC 3 I=1,4
CC J=1,4

```



```

SUNA=0.
SUNB=0.
SUNC=0.
SUNJ=0.
SUN4 L=1,4
SUNJ=SUMA+C(I,L)*A(L,J)
SUNB=SUMB+C(I,L)*BEE(L,J)
SUNC=SUMC+CMEGL(I,L)*H(L,J)
4 PATA(I,J)=SUMA+UNIT(I,J)
PETA(I,J)=SUMB
PETA(I,J)=SUMC
PETA(I,J)=SUMJ
5 I=1,4
J=1,4
SUNCB=0.
SUNCA=0.
SUNCC=0.
SUNCL=1,4
SUNCB=SUMQB+POTA(I,L)*PBTA(L,J)
SUNCA=SUMQA+POTA(I,L)*PATA(L,J)
SUNCC=SUMQC+POTA(I,L)*C(L,J)
6 CLL(I,J)=SUMQB+PJT(I,J)+CAPL1(I,J)
PTR(I,J)=SUMQA
LGG(I,J)=SUMQC
7 CALL MATINV(DLL,4,PTR,4,DETERM,IPIVOT,INDEX,4,ISCALE)
CCL I=1,4
CCL J=1,4
SUNCE=0.
SUNE=0.
SUN7 L=1,4
SUNCE=SUME-DLL(I,L)*DGG(L,J)
SUNE=SUME-DLL(I,L)*CMEGL(L,J)
CCL(I,J,MN)=DLL(I,J)
CCL(I,J,MN)=SUMD
8 DF(I,J,MN)=SUME
I=1+K*MAX*(MN-1)
9 P(I,J,I,J)=PTR(I,J)
CCL I=1,4
CCL J=1,4
10 NN=IABS(N(MN))
IJ=1+K*MAX*(MN-1)
CCL I=1,4
CCL J=1,4
X(I,I,J)=0.
CCL I=1,4
CCL J=1,4
14 P(I,J,I,J)=0.
IF(NN.GT.3) GO TO 11
IF(NN.GT.2) GO TO 50

```

```

00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270

```



```

C      IN  FMATRX
      CALL EFG(2,MN,ZG,Z2,Z3)
      CALL ABC
      CALL MATINV(A,4,G1,0,DETERM,IPIVGT,INDEX,4,ISCALE)
      CC 501 II=1,4
      DC 501 JJ=1,4
      CL(11,JJ,MN)=0.
      CG(11,JJ,MN)=0.
      CF(11,JJ,MN)=0.
      IF(MN.GT.1) GO TO 12
      IF(MN.GT.0) GO TO 13
      MC=MN
      CL(1,1,MN)=1.
      CL(2,2,MN)=1.
      CL(3,3,MN)=-3.
      CL(4,4,MN)=-3.
      CG(3,3,MN)=4.
      DG(4,4,MN)=4.
      CF(3,3,MN)=-1.
      CF(4,4,MN)=-1.
      CC TO SC2
      13 M1=MN
      CL(1,1,MN)=-3.
      CL(2,2,MN)=1.
      CL(2,2,MN)=1.
      IF(M1) .LT.0) DL(2,2,MN)=-1.
      CL(3,3,MN)=1.
      CL(4,4,MN)=1.
      CG(1,1,MN)=4.
      CF(1,1,MN)=-1.
      CC TO SC2
      12 M2=MN
      CL(1,1,MN)=1.
      CL(2,2,MN)=1.
      CL(3,3,MN)=1.
      CL(4,4,MN)=-3.
      CG(4,4,MN)=4.
      CF(4,4,MN)=-1.
      CC INUE
      DC 503 II=1,4
      DC 503 JJ=1,4
      TTF=0.
      CC 504 L=1,4
      TTF=ITP+CF(II,L,MN)*A(L,JJ)
      9C4 CC 505 II=1,4
      9C3 CC 505 JJ=1,4
      TTF=0.

```



```

ITC=0.
CC SC6 L=1,4
ITF=ITP+CLO(II,L)*C(L,JJ)
TTQ=TTQ+CLO(II,L)*BEE(L,JJ)
CL1(II,JJ)=DL(II,JJ,MN)-ITP
CL2(II,JJ)=DG(II,JJ,MN)-TTQ
CALL MATINV(CLL,4,GI,0,DETERM,IPIVOT,INDEX,4,ISCALE)
CC SC7 II=1,4
CC SC7 JJ=1,4
ITP=0.
ITC=C.
CC SC8 L=1,4
ITF=ITP+CCL(II,L)*CLO(L,JJ)
TTQ=TTQ+CCL(II,L)*CL2(L,JJ)
CL(II,JJ,MN)=-ITP
P(II,JJ,IJ)=TTQ
GC TO 11
N2=MN
CCNTI=KMAX
KLAST=KMAX
IF(IBCFL.LT.0) KLAST=KL
CC 23 K=2,KLAST
CC 23 MN=MNINIT,MNMAX
CALL EFG(K,MN,ZO,Z2,Z3)
CALL ABC
CALL FANDD(K,MN,P,CEE,DST,X)
IF(IBCFL.LT.0) GC TO 30
DC 40 MN=MNINIT,MNMAX
IKL=MN*KMAX-I
JKL=KMAX*MN
CALL FJ(KMAX,MN)
CC 41 I=1,4
CC 41 J=1,4
SUMC=0.
SUMF=0.
SUMJ=0.
DC 42 L=1,4
SUMC=SUMC+CMGL(I,L)*H(L,J)
SUMF=SUMF+P(I,L)*F(L,J,JKL)
SUMJ=SUMJ+CMGL(I,L)*JAY(L,J)
PATA(I,J)=SUMO
PETA(I,J)=UNIT(I,J)-SUMF
PETA(I,J)=SUMJ+CAPLL(I,J)
CC 43 I=1,4
CC 43 J=1,4
SUMCP=0.
SUMJP=0.
SUMCM=0.

```

```

00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720

```



```

44      L=1,4
      SUMOP=SUMOP+PATA(I,L)*PBTA(L,J)
      SUMJP=SUMJP+PJTA(I,L)*P(L,J,JKL)
      SUMCN=SUMQM-PATA(I,L)*P(L,J,IKL)
42      ZF1(I,J)=SUMQM-SUMJP
      ZF2(I,J)=SUMQM-PJTA(I,J)
      CALL MATINV(ZF1,4,ZF2,4,DETERM,IFIVOT,INDEX,4,ISCALE)
      L=1,4
      I=1,4
      J=1,4
      ZF3=0.
      ZF4=0.
      L=1,4
      ZF3=SZF3+ZF1(I,L)*PATA(L,J)
      ZF4=SZF4-ZF1(I,L)*CMEGL(L,J)
46      ZF3M(I,J,MN)=SZF3
      ZF4M(I,J,MN)=SZF4
      ZF1M(I,J,MN)=ZF1(I,J)
      ZF2M(I,J,MN)=ZF2(I,J)
45      CCNTINUE
4C      RETURN
      DC=1 MN=MNINIT,MNMAX
      IKL=MN*KMAX-1
      NN=N(MN)
      IF(NN.GT.3) GO TO 31
      IF(NN.GT.2) GO TO 300
      IF(NN.GT.1) GO TO 33
      IF(NN.GT.0) GC TO 34
      MC=MN
      L=1,4
      I=1,4
      J=1,4
      CLC(I,J)=0.
      ZFFC(I,I)=1.
      ZFFC(1,2)=1.
      ZFFC(3,1)=P(3,1,IKL)
      ZFFC(3,2)=P(3,2,IKL)+1.
      ZFFC(3,3)=P(3,3,IKL)
      ZFFC(3,4)=P(3,4,IKL)
      ZFFC(4,1)=P(4,1,IKL)
      ZFFC(4,2)=P(4,2,IKL)
      ZFFC(4,3)=P(4,3,IKL)
      ZFFC(4,4)=P(4,4,IKL)+1.
      CLC(3,3)=1.
      CLC(4,4)=1.
      CALL MATINV(ZFPO,4,CLO,4,DETERM,IFIVOT,INDEX,4,ISCALE)
      GC TO 31
      NC=MN
3C      N2=MN

```

```

0000273C
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200

```



```

34 GC TO 31
   M1=MN J=1,4
   CC 60 I=1,4
   CC 60 J=1,4
   CL1(I,J)=0.
   ZFP1(I,J)=P(1,1,IKL)+1.
   ZFF1(1,1)=P(1,2,IKL)
   ZFF1(1,2)=P(1,3,IKL)
   ZFF1(1,3)=P(1,4,IKL)
   ZFF1(1,4)=P(1,1,IKL)
   ZFF1(2,1)=1.
   ZFF1(2,2)=-1.
   IF(N(MN).LT.0) ZFP1(2,2) = 1.
   ZFP1(3,3)=1.
   ZFP1(4,4)=1.
   CL1(1,1)=1.
   CALL MATINV(ZFP1,4,CL1,4,DETERM,IPIVOT,INDEX,4,ISCALE)
33 GC TO 31
   M2=MN J=1,4
   CC 70 I=1,4
   CC 70 J=1,4
   CL2(I,J)=0.
   ZFP2(1,1)=1.
   ZFP2(2,2)=1.
   ZFP2(3,3)=1.
   ZFP2(4,1)=P(4,1,IKL)
   ZFP2(4,2)=P(4,2,IKL)
   ZFP2(4,3)=P(4,3,IKL)
   ZFP2(4,4)=P(4,4,IKL)+1.
   CL2(4,4)=1.
   CALL MATINV(ZFP2,4,CL2,4,DETERM,IPIVOT,INDEX,4,ISCALE)
31 CC RETURN
   ENCL
   SLERCUTINE FORCE (K,F,X,CEE,DST,2,ZC,Z2,Z3)
   ***** THIS SUBROUTINE COMPUTES THE GEE VECTOR IN EQUATION (28), AND
   ***** THE X VECTOR IN EQUATION (29A) FOR A GIVEN MERICICNAL STATION
   ***** K. ***** THE VECTOR GEES IS THE NON-LINEAR VALUE OF GEE AT STA. 1. *****
   ***** IMPLICIT LOGICAL*1 ($) *****
   REAL AU,MT,LAM2,MAS
   DIMENSION P(4,4,1),CEE(4,4,1),Z(4,1),ZC(4,1),
1 Z2(4,1),Z3(4,1),X(4,1)
   COMMON /IBL1/ MNMAX
   COMMON /IBL2/ N(95),MNINIT
   COMMON /IBL4/ KMAX,KL
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680

```



```

CCMCMCN /IBL5/ IBCINL, IBCFNL
CCMCMCN /IBL8/ LSTEP, ITR
CCMCMCN /IBL12/ KMAX1, KMAX2, NCONV
CCMCMCN /IBL13/ ITRMAX, LSMAX
CCMCMCN /BL3/ PR(99), PX(99), PT(99)
CCMCMCN /BL4/ ZF1M(4,4,99), ZF2M(4,4,99),
1 ZF3M(4,4,99), ZF4M(4,4,99)
CCMCMCN /BL5/ TT(99), MT(99), DT(99), DMT(99)
CCMCMCN /BL6/ SQE, GSE, ALOAD
CCMCMCN /BL7/ D1, S1
CCMCMCN /BL8/ R(500), GAM(500), DMT(500)
CCMCMCN /BL9/ FFS(4,99), ELIS(4), GEES(4,99)
CCMCMCN /BL11/ OMXI(500), PHEE, TO, T2
CCMCMCN /BL12/ TOLI, TDEL
CCMCMCN /BL14/ LAM2, LSD18, LSD1N
CCMCMCN /BL15/ NU, UI(99), VI(99), W1(99), V2(99), U2(99), W2(99), U3(99),
1 V3(99), W3(99)
CCMCMCN /BL17/ DEL
CCMCMCN /BL24/ DL(4,4,99), DG(4,4,99), DF(4,4,99)
CCMCMCN /BL27/ BX3(99), BT3(99), BX13(99), BE3(99)
CCMCMCN /BL28/ EXX3(99), ET13(99), ETX3(99), EX13(99), ET3(99)
CCMCMCN /BL29/ BX1(99), BT1(99), BX11(99), BE1(99), BX2(99), BT2(99),
1 BX12(99), BE2(99)
CCMCMCN /BL30/ EXX1(99), ET11(99), ETX1(99), EX1(99), EX2(99),
1 ET12(99), ETX2(99), EX2(99), ET2(99)
CCMCMCN /BL31/ DELSG, EX11(99)
CCMCMCN /BL100/ TEEQ, $DYNMC
CCMCMCN /BL101/ DELSO
CCMCMCN /BL102/ DELSO
CCMCMCN /BL103/ MASS(500)
DIMEASICN GEE(4)
C*****A+2.*B-.5*C)/CEL
FCIFF(A,B,C)=(-1.5*A+2.*B-.5*C)/CEL
RS=R(K)
RF=1./RS
GA=GAM(K)
CX=CMXI(K)
CT=CMT(K)
CL2=D1*LAM2
CALL BCB(K,BS,DBS,D,DD)
CALL PLCAD(K,Z)
CALL PLCAD(K,Z)
MAS=MASS(K)
CC 4 A=1,MNMAX
I2=K+1+(M-1)*KMAX2
IK=K+(M-1)*KMAX
IK1=IK-1
EN=N(M)
00003690
00003700
00003710
00003720

00003750
00003760
00003770
00003780

00003800
00003810
00003820
00003830
00003840
00003850

00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004150

```



```

ENR=EN*RR
ENT=MT(M)
GEE(1)=(-PX(M)+DT(M)-DL2*GA*QX*EMT)*TDEL*ALCAD
1 GEE(2)=(-5.*ZD(1,IZ)+4.*Z2(1,IZ)-Z3(1,IZ))*TDEL/DELS
1 GEE(3)=(-PT(M)-ENR*TT(M)-CL2*ENR*QT*EMT)*TDEL*ALCAC
1 GEE(4)=(-5.*ZD(2,IZ)+4.*Z2(2,IZ)-Z3(2,IZ))*TDEL/DELS
1 GEE(5)=(-PR(M)-(CX+CT)*TT(M)-CL2*(GA*CMT(M)-(CX*CT-ENR**2)*TDEL*ALCAD
1 GEE(6)=(-5.*ZD(3,IZ)+4.*Z2(3,IZ)-Z3(3,IZ))*TDEL/DELS
1 GEE(7)=MT(M)*TDEL*ALCAD
IF(IJTRMAX.EQ.1) GO TO 5C
IF(K.GT.1) GO TO 6
BX2T=BX1(M)
BX12T=BX11(M)
BE2T=BE1(M)
EXX2T=EXX1(M)
ET12T=ET11(M)
EX12T=EX11(M)
ET2T=ET1(M)
CET=FCIFF(BX2T,BX3(M))
CET=FCIFF(BT2T,BT3(M))
CET=FCIFF(BX12T,BX13(M))
CEE=FCIFF(BE2T,BE3(M))
CEX=FCIFF(ET2T,ET3(M))
DEXX=FCIFF(EX2T,EX3(M))
CETX=FCIFF(ETX2T,ETX3(M))
GO TO 7
IF(K.LT.KMAX) GO TC 8
BX2T=BX3(M)
BT2T=BT3(M)
BX12T=BX13(M)
BE2T=BE3(M)
EXX2T=EXX3(M)
ET12T=ET13(M)
EX12T=EX13(M)
ET2T=ET3(M)
CET=FCIFF(BX2T,BX1(M))
CET=FCIFF(BT2T,BT1(M))
CET=FCIFF(BX12T,BX11(M))
CEE=FCIFF(BE2T,BE1(M))
CEX=FCIFF(ET2T,ET1(M))
DEXX=FCIFF(EX2T,EX1(M))
CETX=FCIFF(ETX2T,ETX1(M))

```



```

C2XX=-FCIFF(EXX2T,EXX2(M),EXX1(M))
C2TX=-FCIFF(ETX2T,ETX2(M),ETX1(M))
C TO 7
C CBX=TDLI*(BX3(M)-BX1(M))
C CBE=TDLI*(BT3(M)-BT1(M))
C CBE=TDLI*(BE3(M)-BE1(M))
C CBXT=TDLI*(BX3(M)-BX1(M))
C CBXT=TDLI*(BT3(M)-BT1(M))
C CBXT=TDLI*(BE3(M)-BE1(M))
C CBXT=TDLI*(EXX3(M)-EXX1(M))
C CBXT=TDLI*(ETX3(M)-ETX1(M))
C CBXT=TDLI*(BX2(M))
C CBXT=TDLI*(BT2(M))
C CBXT=TDLI*(BE2(M))
C CBXT=TDLI*(EXX2(M))
C CBXT=TDLI*(ETX2(M))
C CBXT=TDLI*(EX2(M))
C CBXT=TDLI*(ET2(M))
C GEE(1)=GEE(1)-OSE*(BS*(CBX+DBE+GA*DL*(BX2T-ET2T)+NU*(CBT+DBE)
+ENR*DL*(BX2T+ET2T)+CBS*(BX2T+BE2T+NU*(BT2T+ET2T))*TDEL
(GEXX2T+ETX2T)-ENR*(EXX2T+ETX2T))*TDEL
C GEE(2)=GEE(2)+OSE*(BS*(CBX+DBE+GA*DL*(BX2T-ET2T)+NU*(CBT+DBE)
+ENR*DL*(BX2T+ET2T)+CBS*(BX2T+BE2T+NU*(BT2T+ET2T))*TDEL
(GEXX2T+ETX2T)-ENR*(EXX2T+ETX2T))*TDEL
C GEE(3)=GEE(3)+OSE*(BS*(CBX+DBE+GA*DL*(BX2T-ET2T)+NU*(CBT+DBE)
+ENR*DL*(BX2T+ET2T)+CBS*(BX2T+BE2T+NU*(BT2T+ET2T))*TDEL
(GEXX2T+ETX2T)-ENR*(EXX2T+ETX2T))*TDEL
C IF(K.GT.1) GO TO 10
C IF(M.GT.1) GO TO 10
C CC 20 I=1,4
C GEES(I,M)=GEE(I)
C SUMX=0.
C CC 21 J=1,4
C SUMX=SUMX+GEE(I,M)*ELIS(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)
C FLOWING CARD CAUSES A SPECIFIED BOUNDARY CCNDITCN VALUE TC
C EXISTS CNLY FOR MCDE 0.
C SUMX=SUMX+DL(I,J,M)*ELIS(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)
C IF(M.NE.1) ELIS(J)=0.
C SUMX=SUMX+DL(I,J,M)*ELIS(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)
C X(I,IK)=SUMX
C TO 4
C IN FCRCE
C IF(K.NE.2) OR (K.EQ.2.ANC.IBCINL.GE.0) GC TC 501
C CC 502 I1=1,4
C SUMX=0.

```

```

00004640
00004650
00004660
00004670
00004680
00004690
00004700
00004710
00004720
00004730
00004740
00004750
00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070

```







```

00005520
00005530
00005540
00005550
00005560
00005570
00005580
00005590
00005600
00005610
00005620
00005630
00005640
00005650
00005660
00005670
00005680
00005690
00005700
00005710
00005720
00005730
00005740
00005750
00005760
00005770
00005780
00005790
00005800
00005810
00005820
00005830
00005840
00005850
00005860
00005870
00005880
00005890
00005900
00005910
00005920
00005930
00005940
00005950
00005960
00005970
00005980
00005990

RETURN
END
*****
C ***** MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)
C ***** THIS SUBROUTINE SOLVES THE MATRIX EQUATION AX=B, WHERE A IS
C ***** A SQUARE COEFFICIENT MATRIX AND B IS A MATRIX CF CCNSTANT VEC-
C ***** TORS.
C ***** A(INVERSE) IS ALSO OBTAINED AND THE DETERMINANT CF A IS AVAIL-
C ***** ABLE.
C ***** THE FOLLOWING MUST BE DIMENSIONED IN THE CALLING PROGRAM:
C ***** IPIVOT(N MAX), INDEX(N MAX,2), A(N MAX,N MAX), B(N MAX,N MAX)
C ***** WHERE:
C ***** A = NAME OF 2-DIMENSIONAL ARRAY TO BE INVERTED
C ***** N = ORDER CF A - 1<=N<=NMAX
C ***** B = NAME CF 2-DIMENSIONAL ARRAY TO BE MULTIPLIED
C ***** BY A(INVERSE)
C ***** M = NUMBER CF COLUMNS VECTORS IN B
C ***** NOTE: M = 0 SIGNALS INVERSCN ONLY)
C ***** IPIVOT = TEMPORARY STORAGE BLOCK
C ***** INDEX = TEMPORARY STORAGE BLOCK
C ***** NMAX = MAXIMUM ORDER OF A (AS DIMENSIONED IN THE
C ***** CALLING PROGRAM)
C ***** DETERM= VALUE CF DETERMINANT AS GIVEN BELOW
C ***** ISCALE = USED IN FORMULA BELCW
C ***** DETERMINANT(A) = (1C*18)**ISCALE*(DETERM)
C ***** A(INVERSE) IS STORED IN A
C ***** A(INVERSE)*B IS STORED IN B
C ***** DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)
C ***** EQUIVALENCE (IROW,JRCW), (ICOLU,JCCLU), (I,SWAP)
C ***** INITIALIZATION
C *****
C ***** ISCALE=C**18
C ***** R1=10.0/R1
C ***** R2=1.0/R1
C ***** I=1
C ***** J=1,N
C ***** IF IPIVOT(J)=0
C ***** DC 550 I=I+1,N
C ***** SEARCH FOR PIVOT ELEMENT
C *****
C ***** AMAX=0.C
C ***** CC 105 J=1,N
C ***** 5C IF (IPIVOT(J)-1) 60, 105, 60

```



```

6C CC 100 K=1,N
7C IF (IPIVOT(K)-1) 80, 100, 740
8C IF (ABS(AMAX)-ABS(A(J,K)))85,10C,100
9C IF C=J

```

```

5C ICCLUM=K
5C AMAX=A(J,K)
1CC CONTINUE
1CE CONTINUE
11C IF IVOT(ICCLUM)=IPIVCT(ICCLUM)+1

```

```

C C INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGCNA
C C

```

```

12C IF (IROW-ICCLUM) 140, 260, 140
14C CTERM=-CTERM
15C CC 200 L=1,N
16C SWAP=A(IROW,L)
17C A(IROW,L)=A(ICCLUM,L)
20C A(ICCLUM,L)=SWAP
21C IF(M) 260, 260, 210
22C CC 250 L=1, M
23C SWAP=B(IROW,L)
24C B(IROW,L)=B(ICCLUM,L)
25C B(ICCLUM,L)=SWAP
26C INDEX(I,1)=IROW
27C INDEX(1,2)=ICCLUM
31C PIVCT=A(ICCLUM,ICCLUM)

```

```

C C SCALE THE DETERMINANT
C C

```

```

10CC PIVCTI=PIVOT
10C5 IF(ABS(CTERM)-R1)1030,1010,1010
101C CTERM=CTERM/R1
102C ISCALE=ISCALE+1
102C IF(ABS(CTERM)-R1)1060,1020,102C
102C CTERM=CTERM/R1
102C ISCALE=ISCALE+1
103C GC TO 1060
104C IF(ABS(CTERM)-R2)1040,1040,1060
104C CTERM=CTERM*R1
105C ISCALE=ISCALE-1
105C IF(ABS(CTERM)-R2)1050,1050,1060
105C CTERM=CTERM*R1
106C ISCALE=ISCALE-1
106C IF(ABS(PIVCTI)-R1)109C,1070,107C
107C PIVCTI=PIVCTI/R1
107C ISCALE=ISCALE+1
108C IF(ABS(PIVCTI)-R1)320,1080,1080
108C PIVCTI=PIVCTI/R1

```

```

00C06C00
00006010
00006020
00006030
00006C40
00006050
00C06060
00006070
00C06080
00006090
00C06100
00006110
00006120
00006130
00C06140
00C06150
00006160
00006170
00006180
00006190
00C06200
00006210
00006220
00006230
00006240
00006250
00006260
00006270
00C06280
00006290
00006300
00006310
00006320
00C06330
00006340
00006350
00006360
00006370
00006380
00006390
00006400
00C06410
00006420
00006430
00006440
00C06450
00006460
00006470

```











```

BX1(M2)=BET
BT1(M2)=-BET
EXT1(M2)=-BET
EXT1(M1)=Q1
IF(M3.EQ.0) GO TO 2
EXT1(M3)=Q1
EXT1(M3)=-Q1
2 TC=0.
IF(MO.EQ.0) GO TO 3
EXT1(MO)=BET
BT1(MO)=BET
CALL BCE(1,B,DB,D,CC)
CALL TLCD(1,Z)
I2=2+(MO-1)*KMAX2
I3=I2+1
I4=I3+1
TC=B*SI*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+CMXI(1)*Z(3,I2)
1+.5*SOE*BET)-TT(MO)*ALCD
3 EXT1(M1)=PEE*(TO+.5*T2)
RETURN
10CC CCNTINUE
PFEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+CMXI(1)*Z(1,I2)
T2=C.
IF(M2.EQ.0) GO TO 1002
CALL BDB(1,B,DB,D,CC)
I2=2+(M2-1)*KMAX2
I3=I2+1
I4=I3+1
PTX1=PHIXB(KMAX+1)
PTX2=PHIXB(2*KMAX+1)
PFEN=(1.5*Z(3,I2-KMAX2)-2.*Z(3,I3-KMAX2)+.5*Z(3,I4-KMAX2))/DEL+
1CMXI(1)*Z(1,I2-KMAX2)
BX1(M2)=.5*(PEE*(PTX1)-PTEN*(PHEN+2.*PHX2))
IF(ITRMAX.EQ.1) BX1(M2)=0.
PT1(M2)=-BX1(M2)
EXT1(M2)=-BX1(M2)
T2=B*DI*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BX1(M2))
M2L=M2-1
EXT1(M2L)=PEE*(PTEN+PHX2)+PHX1*PTEN
IF(ITRMAX.EQ.1) BX1(M2L)=0.
PT1(M2L)=-BX1(M2L)
EXT1(M2L)=BX1(M2L)
T2A=B*CI*((-1.5*Z(1,I2-KMAX2)+2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))
1/DEL+.5*SOE*BX1(M2L))
1002 TC=C.
IF(MO.EQ.0) GO TO 1003
EXT1(MO)=.5*(PEE*(PTX1)+PTEN*(PHEN+2.*PHX2))
IF(ITRMAX.EQ.1) BX1(MO)=0.

```







```

C P-HAT MATRICES FOR EACH MEDICIAN STATION K AND FOURIER MCCE MN*00008400
C THESE MATRICES ARE COMPUTED AND SAVED BECAUSE THEY DC NCT *00008410
C CHANGE DURING EITHER THE ITERATION PROCEDURE OR THE LOAC INCRE-00008420
C WENT PROCEDURE - AS THEY ARE A FUNCTION CF THE SPELL'S INITIAL*00008430
C GEOMETRY AND STIFFNESS. *00008440
C *** *00008450
C DIMENSION P(4,4,1),CEE(4,4,1),DST(4,4,1),X(4,1) *00008460
C COMMON /IBL4/ KMAX, KL *00008470
C COMMON /BL1/ A(4,4,99),ZF2M(4,4,99), *00008480
C COMMON /BL4/ ZF1M(4,4,99),ZF4M(4,4,99), *00008490
C *** *00008500
C 1 DIMENSION TM(4,4),IPIVCT(4),INDEX(4,2),X2(4) *00008510
C *** *00008520
C IKL=K+KMAX*(MN-1) *00008530
C KLI=IKL-1 *00008540
C DO 1 I=1,4 *00008550
C DO 1 J=1,4 *00008560
C SUM=0. *00008570
C DO 2 L=1,4 *00008580
C SUM=SUM+C(I,L)*P(L,J,KLI) *00008590
C TM(I,J)=BEE(I,J)-SUM *00008600
C 1 CALL MATINV(TM,4,X2,0,DETERM,IPIVCT,INDEX,4,ISCALE) *00008610
C DO 5 I=1,4 *00008620
C DO 5 J=1,4 *00008630
C SUMA=0. *00008640
C DO 6 L=1,4 *00008650
C SUMA=SUMA+TM(I,L)*A(L,J) *00008660
C SUMC=SUMC+TM(I,L)*C(L,J) *00008670
C 6 P(I,J,IKL)=SUMA *00008680
C DEE(I,J,IKL)=TM(I,J) *00008690
C 5 DCT(I,J,IKL)=SUMC *00008700
C 5 RETURN *00008710
C ENCL *00008720
C SUBROUTINE FNLPOOL (Z,PHIXB,PHITB) *00008730
C *** *00008740
C THIS SUBROUTINE CCMPUTES THE NCN-LINEAR TERMS BETA-SUB S, *00008750
C -SUB THETA, -SUB S-THETA, ETA-SUB S-S, AND -SUB THETA-S AT A *00008760
C FINAL PCLE. *00008770
C *** *00008780
C DIMENSION Z(4,1),PHIXB(1),PHITB(1) *00008790
C COMMON /IBL1/ MNMAX *00008800
C COMMON /IBL3/ M0,M1,M2,M3 *00008810
C COMMON /IBL4/ KMAX, KL *00008820
C COMMON /IBL12/ KMAX1,KMAX2,NCONV *00008830
C COMMON /IBL13/ ITRMAX,LSMAX *00008840
C COMMON /IBLJ/ JUMP *00008850
C COMMON /BL5/ TT(99),EMT(99),DT(99),DMT(99) *00008860
C *** *00008870

```



```

00008880
00008890
00008900
00008910
00008920
00008930
00008940
00008950
00008960
00008970
00008980
00008990
00009000
00009010
00009020
00009030
00009040
00009050
00009060
00009070
00009080
00009090
00009100
00009110
00009120
00009130
00009140
00009150
00009160
00009170
00009180
00009190
00009200
00009210
00009220
00009230
00009240
00009250
00009260
00009270
00009280
00009290
00009300
00009310
00009320
00009330
00009340
00009350
CCMCON /BL6/ SOE,CSE,ALOAD
CCMCON /BL7/ D1,$1
CCMCON /BL11/ OMXI(500),PHEE,TO,T2
CCMCON /BL11A/ PHEN,T2N
CCMCON /BL17/ DEL
CCMCON /BL27/ BX3(59),BT3(99),BX13(99),BE2(59)
CCMCON /BL28/ EXX3(59),ET13(99),ETX3(99),EX3(99),ET3(99)
C*****
DC 1 MN=1,MNMAX
EX3( MN)=0.
ET3( MN)=0.
BX13( MN)=C.
EX3( MN)=0.
ET3( MN)=0.
ETX3( MN)=C.
EXX3( MN)=C.
CALL BCB(KMAX,B,DB,D,DB)
IF(M1.EC.O) RETURN
KM=KMAX1+(M1-1)*KMAX2
KM1=KM-1
KM2=KM-2
PFEE=-(1.5*Z(3,KM1)+.5*Z(3,KM2))/DEL+CMXI(KMAX)*Z(1,KM)
IF(JUMP.EQ.2) GO TO 1000
BET=.5*PHEE#2
IF(ITRMAX.EQ.1) BET=0.
T2=C.
IF(M2.EQ.0) GO TO 2
KA=KMAX1+(M2-1)*KMAX2
KA1=KM-1
KA2=KM-2
T2=B*DL1*((1.5*Z(1,KM)-2.*Z(1,KM1))+.5*Z(1,KM2))/DEL+.5*SCE*BET)
C1=.5*PFEE#T2
EX3( M2)=BET
ET13( M2)=-BET
ETX3( M1)=Q1
IF(M3.EQ.0) GO TO 2
EXX3( M3)=Q1
ETX3( M3)=-Q1
TC=0.
IF(M0.EQ.0) GO TO 3
CALL TLCD(KMAX,Z)
KA=KMAX1+(M0-1)*KMAX2
KA1=KM-1
KA2=KM-2
EXX3( M0)=BET
ET3( M0)=BET

```



```

TC=B*SI*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+CMXI(KMAX))*
12(3,KM)+.5*SDE*BEI)-TI(M0)*ALOAD
3 EXX3(M1)=PHEE*(TO+.5*TI)
C*****
10CC CCNTINUE
T2=C.EQ.0) GO TO 1002
IF(M2.EQ.0) GO TO 1002
KM=KMAX1+(M2-1)*KMAX2
KA1=KM-1
KA2=KM-2
J=KMAX*2
I=J+KMAX
PHX1=PHIXB(J)
PHX2=PHIXB(I)
PHEN=-(1.5*Z(3,KM-KMAX2)-2.*Z(3,KM1-KMAX2)+.5*Z(3,KM2-KMAX2))/DEL
1+CMXI(KMAX)*Z(1,KM-KMAX2)
BX3(M2)=.5*(PHEE*(PHEN+2.*PHX1)-PHEN*(PHEN+2.*PHX2))
IF(ITRMAX.EQ.1) BX3(M2)=0.
ET3(M2)=-BX3(M2)
BX13(M2)=BX3(M2)
M2L=M2-1
BX3(M2L)=PHEE*(PHEN+PHX2)+PHX1*PHEN
IF(ITRMAX.EQ.1) BX3(M2L)=0.
ET3(M2L)=-BX3(M2L)
EX13(M2L)=-BX3(M2L)
T2=B*SI*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SCE*BX3(M2))
T2A=B*SI*((1.5*Z(1,KM-KMAX2)-2.*Z(1,KM1-KMAX2)+.5*Z(1,KM2-KMAX2))
1/DEL+.5*SDE*BX3(M2L))
10C2 TC=C.
IF(M0.EQ.0) GO TO 1003
CALL TLCD(KMAX,2)
KM=KMAX1+(M0-1)*KMAX2
KA1=KM-1
KA2=KM-2
BX3(M0)=.5*(PHEE*(PHEE+2.*PHX1)+PHEN*(PHEN+2.*PHX2))
IF(ITRMAX.EQ.1) BX3(M0)=0.
ET3(M0)=BX3(M0)
TC=B*SI*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+CMXI(KMAX))*
12(3,KM)+.5*SDE*BEI)-TI(M0)*ALOAD
10C3 IF(ITRMAX.EQ.1) GO TO 1001
PHSS=PHEN+PHX1
PHSP=PHEN+PHX2
M1L=M1-1
EXX3(M1)=PHSS*TO+.5*(PHSS*TI2+PHSP*TI2N)
EXX3(M1L)=PHSP*TO-.5*(PHSP*TI2-PHSS*TI2N)
ETX3(M1)=.5*(PHSS*TI2+PHSP*TI2N)
ETX3(M1L)=.5*(-PHSP*TI2+PHSS*TI2N)

```



```

00009840
00009850
00009860
00009870
00009880
00009890
00009900
00009910
00009920
00009930
00009940
00009950
00009960
00009970
00009980
00009990
00010000
00010010
00010020
00010030
00010040
00010050
00010060
00010070
00010080
00010090
00010100
00010110
00010120
00010130
00010140
00010150
00010160
00010170
00010180
00010190
00010200
00010210
00010220
00010230
00010240
00010250
00010260
00010270
00010280
00010290
00010300
00010310

IF(M3.EQ.0) GO TO 1001
M3L=M3-1
EXX3=(M3L)=.5*(PHSS*T2-PHSP*T2N)
EXX3=(M3L)=.5*(PHSS*T2N+PHSP*T2)
ETX3=(M3L)=.5*(-PHSS*T2+PHSP*T2N)
ETX3=(M3L)=.5*(-PHSS*T2-PHSS*T2N)
1001 CCNTINUE
RETURN
END
SUBROUTINE PHIBET(K,Z,IS,JS,ID,JC,PHIXB,PHITB)
*****
C THIS SUBROUTINE CALCULATES THE PHI'S AND CARRIES CLT THE BETA
C MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE BETA
C NCN-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K. THE ARRAYS
C IS, ID, JS, IJS, MAXS, MAXC, MAXSY ARE PREPARED IN SUB-
C ROUTINE MODES AND USED HERE.
*****
C DIMENSION Z(4,1), IS(99,1), JS(99,1), ID(99,1), JD(99,1), PHIXB(1),
C PHITB(1)
C COMMON /IBL1/ MNMAX
C COMMON /IBL2/ N(99), MNINIT
C COMMON /IBL4/ KMAX, KL
C COMMON /IBL7/ MNMAXO, MAXD(99), MAXS(59), MAXSY(55), IJS(59)
C COMMON /IBL12/ KMAX1, KMAX2, NCONV
C COMMON /IBL13/ ITRMAX, LSMAX
C COMMON /IBLJ/ JUMP
C COMMON /IBL6/ SOE, CSE, ALGAD
C COMMON /BL8/ R(500), GAM(500), OMT(500)
C COMMON /BL10/ PHIX(99), PHIT(99), PHI(99)
C COMMON /BL11/ OMXI(500), PHEE, TO, T2
C COMMON /BL12/ TDLI, TDEL
C COMMON /BL15/ NU, UI(99), V1(99), V2(59), U2(99), W2(59), U3(59),
C V3(99), W3(99)
C COMMON /BL27/ BX3(59), BT3(99), BXT3(99), BE3(55)
C COMMON /BLPHS/ PHX(99), PHT(99)
*****
C X=CMXI(K)
C T=CMT(K)
C RA=1./R(K)
C GA=GAM(K)
C FZ=K+2
C C 1 N=1, MNMAXO
C EN=A(M)
C IK=KP2+(M-1)*KMAX2
C U3(M)=Z(1,IK)
C V3(M)=Z(2,IK)
C W3(M)=Z(3,IK)
C PHIX(M)=-TDLI*(W3(M)-W1(M))+CX*L2(M)
*****

```



```

1 PHIT(M)=EN*W2(M)*RRRA+V2(M)*CT
  PHI(M)=(TDLI*(V3(M)-VI(M))+GA*V2(M)+EN*U2(M)*RRRA)*.5
  IF(ISTRMAX.EQ.1) RETURN
  IF(JUMP.EQ.2) GO TC 1111
  CC 5 N=1,MNMAX
  SMC=0.
  SMT=0.
  SMF=0.
  IF(N(M).EQ.0) GO TC 20
  MAXL=MAXS(M)
  IF(MAXL.EQ.0) GO TC 2
  CC 3 L=1,MAXL
  I=IS(L,N)
  J=JS(L,N)
  SMT=SMQ+PHIX(I)*PHIX(J)
  SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
  SMF=SMF+PHI(I)*PHI(J)
  MAXL=MAXL-1
  IF(MAXL.EQ.0) GO TC 4
  CC 5 L=1,MAXL
  I=IC(L,N)
  J=JC(L,N)
  SMT=SMQ+PHIX(I)*PHIX(J)
  SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
  SMF=SMF+PHI(I)*PHI(J)
  IF(MAXSY(M).EQ.0) GC TO 10
  I=IJS(M)
  SMC=SMQ+PHIX(I)**2/2.
  SMT=SMT+PHIT(I)**2/2.
  SMR=(SMR+PHIX(I)*PHIT(I))
  SMF=SMF+PHI(I)**2/2.
  GC TO 1C
  CC 21 L=1,MNMAXO
  SMC=SMQ+PHIX(L)**2
  SMT=SMT+PHIT(L)**2
  SMF=SMF+PHI(L)**2
  IF(N.GT.MNMAXC) GC TO 11
  SMC=SMQ+PHIX(M)**2
  SMT=SMT+SMC*.5
  SMF=SMF+SMF*.5
  IF(T3(M)=0.
  GC TO 9
  EX3(N)=SMT
  BT3(N)=SMQ

```

```

00010320
00010330
00010340
00010350
00010360
00010370
00010380
00010390
00010400
00010410
00010420
00010430
00010440
00010450
00010460
00010470
00010480
00010490
00010500
00010510
00010520
00010530
00010540
00010550
00010560
00010570
00010580
00010590
00010600
00010610
00010620
00010630
00010640
00010650
00010660
00010670
00010680
00010690
00010700
00010710
00010720
00010730
00010740
00010750
00010760
00010770
00010780
00010790

```







```

C**      CCMPLE SUMS FOR SYMMETRIC SUMMATION COMBINATION MCDCE
C**      SFC=SMC+PHX(I)*PHX(J)+PHX(J)*PHX(I)
C**      1 -PHX(I)*PHX(J)+PHX(J)*PHX(I)-PHX(JJ)*PHX(II)
C**      2 SMT=SMT-PHT(I)*PHT(J)-PHT(J)*PHT(I)
C**      1 -PHT(I)*PHT(J)+PHT(J)*PHT(I)
C**      2 SRR=SMR+PHX(I)*PHX(J)+PHX(J)*PHX(I)+PHT(JJ)*PHT(II)
C**      1 +PHX(I)*PHT(J)+PHT(J)*PHT(I)+PHX(I)*PHT(J)
C**      2 3 +PHX(J)*PHT(I)
C**      4 +PHT(II)*PHT(JJ)+PHT(JJ)*PHT(II)
C**      1 SRF=SMF-PHI(I)*PHI(J)
C**      2 +PHI(I)*PHI(J)
C**      3 +PHI(II)*PHI(JJ)+PHX(JJ)*PHT(II)
C**      4 +PHT(II)*PHI(J)
C**      1 CCNTINUE
C**      43 MAXL=MAXD(M)
C**      42 TEST FOR PRESENCE CF SYMMETRIC DIFFERENCE COMBINATIONS
C**      1 IF(MAXL.EQ.0) GO TO 44
C**      2 SET UP -COUPLING- MODES- INDICES AND TEST FOR MODE 1
C**      3 DC 45 L=1,MAXL
C**      4 J=JC(L,M)
C**      1 I=I-1
C**      2 JJ=J-1
C**      3 IF (J.EQ.1) GO TO 442
C**      4 CCMPLE SUMS FOR SYMMETRIC DIFFERENCE COMBINATION MCDCE
C**      1 SFC=SMC+PHX(I)*PHX(J)+PHX(J)*PHX(I)
C**      2 +PHX(I)*PHX(J)+PHX(JJ)+PHX(JJ)*PHX(II)
C**      1 SMT=SMT+PHT(I)*PHT(J)+PHT(J)*PHT(I)
C**      2 +PHT(I)*PHT(J)+PHT(JJ)+PHT(JJ)*PHT(II)
C**      1 SRR=SMR+PHX(I)*PHX(J)+PHX(J)*PHX(I)+PHT(JJ)*PHT(II)
C**      2 -PHX(I)*PHT(J)+PHT(J)*PHT(I)+PHX(I)*PHT(J)
C**      3 +PHX(J)*PHT(I)
C**      4 +PHT(II)*PHT(JJ)+PHT(JJ)*PHT(II)
C**      1 SRF=SMF+PHI(I)*PHI(J)
C**      2 +PHI(I)*PHI(J)
C**      3 +PHI(II)*PHI(JJ)+PHX(JJ)*PHT(II)
C**      4 +PHT(II)*PHI(J)
C**      1 GC TO 45
C**      43

```



```

C ** EXECUTE BELOW IF J=1 IN DIFFERENCE COMBINATIONS **00011760
C ** 442 SMC=(PHIX(1)*PHIX(1)+PHX(1)*PHX(1)+PHIX(1)*PHIX(1))*2.0 *00011770
C ** SAT=SMT+(PHIT(1)*PHIT(1)+PHT(1)*PHT(1)+PHIT(1)*PHT(1))*2.0 *00011780
C ** SMR=SMR+(PHIX(1)*PHIX(1)+PHX(1)*PHX(1)+PHIT(1)*PHT(1))*2.0 *00011800
C ** 1 SMF=SMF+(PHIT(1)*PHIT(1)+PHT(1)*PHT(1)+PHIX(1)*PHIX(1))*2.0 *00011810
C ** 45 CCONTINUE *00011820
C ** TEST FOR PRESENCE OF SAME-INDEX COMBINATION *00011830
C ** 44 IF(MAXSY(M).EQ.0) GO TO 410 *00011840
C ** SET UP INDICES AND COMPILE SUMS *00011850
C ** I=IJS(M) *00011860
C ** II=I-1 *00011870
C ** SMC=SMC+PHIX(1)*PHIX(1)*2/2. *00011880
C ** 1 -PHIX(1)*PHIX(1)*2/2. *00011890
C ** 2 SMT=SMT-PHT(1)*PHT(1)*2/2. *00011900
C ** 1 +PHIT(1)*PHIT(1)*2/2. *00011910
C ** 2 SMR=SMR+PHIX(1)*PHIX(1)*2/2. *00011920
C ** 1 +PHIT(1)*PHIT(1)*2/2. *00011930
C ** 2 SMR=SMR+PHIX(1)*PHIX(1)*2/2. *00011940
C ** 1 +PHIT(1)*PHIT(1)*2/2. *00011950
C ** 2 SMR=SMR+PHIX(1)*PHIX(1)*2/2. *00011960
C ** 1 +PHIT(1)*PHIT(1)*2/2. *00011970
C ** 2 SMF=SMF-PHI(1)*PHI(1)*2/2. *00011980
C ** 1 +PHIX(1)*PHIX(1)*2/2. *00011990
C ** 2 SMF=SMF-PHI(1)*PHI(1)*2/2. *00012000
C ** GC TO 410 *00012010
C ** THIS SECTION HANDLES CASES WHERE MODE 0 IS INCLUDED *00012020
C ** 42C CC 421 L=1,MNMAXC *00012030
C ** SMC=SMC+PHIX(L)*PHIX(L) *00012040
C ** 1 SMT=SMT+PHIT(L)*PHIT(L) *00012050
C ** 1 SMF=SMF+PHI(L)*PHI(L) *00012060
C ** 421 SMC=SMC+PHIX(M)*PHIX(M) *00012070
C ** 1 SMF=SMF+PHI(M)*PHI(M) *00012080
C ** SMT=SMT+PHIT(1)*PHIT(1)+2.0*PHIT(1)*PHIT(1)*2.0 *00012090
C ** SMR=SMR+(PHIX(1)*PHIX(1)+PHX(1)*PHX(1)+PHIT(1)*PHT(1))*2.0 *00012100
C ** CC 122 L=2,MNMAXC *00012110
C ** SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L+1)+PHT(L+1)+PHX(L)*PHX(L) *00012120
C ** 122 SMC=SMC+PHIX(L)*PHIX(L)+PHIT(L+1)+PHT(L+1)+PHX(L)*PHX(L) *00012130
C ** 121 SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012140
C ** 121 SMR=SMR+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012150
C ** CCONTINUE *00012160
C ** 121 SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012170
C ** 121 SMR=SMR+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012180
C ** 121 SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012190
C ** 121 SMR=SMR+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012200
C ** 121 SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012210
C ** 121 SMR=SMR+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012220
C ** 121 SMF=SMF+PHIX(L)*PHIX(L)+PHIT(L-1)+PHT(L-1)+PHX(L)*PHX(L) *00012230

```







```

DC 105 L=1,MAXL
I=JC(L,MP)
J=JC(L,MP)
I=I-1
J=J-1
C** TEST FOR MODE 1 AND CCMPLE SUMS
C** IF (J.EC.1) GO TO 123
C** SMC=SMO-PHX(I)*(PHX(JJ)+PHX(JJ))*PHX(JJ)*(PHX(JJ)+
1 SMC=SMO-PHX(I)+PHX(JJ)*PHX(JJ)-PHX(JJ)*PHX(JJ)+
1 SMC=SMT+PHX(I)*(PHX(JJ)+PHX(JJ))*PHX(JJ)*(PHX(JJ)+
1 SMC=SMT+PHX(I)-PHX(JJ)*PHX(JJ)+PHX(JJ)*PHX(JJ)+
2 SMC=SMT+PHX(I)+PHX(JJ)*PHX(JJ)+PHX(JJ)*PHX(JJ)+
3 SMC=SMT+PHX(I)+PHX(JJ)*PHX(JJ)+PHX(JJ)*PHX(JJ)+
SMF=SMF+PHX(I)*PHX(JJ)-PHX(JJ)*PHX(JJ)
GC TO 1C5
C** EXECUTE BELOW IF J=1 IN DIFFERENCE COMBINATIONS
C** SMC=SMT+PHX(I)*(PHX(JJ)+PHX(JJ))*PHX(JJ)*(PHX(JJ)+
123 SMC=SMT+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
SMC=SMT+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
1 SMC=SMT+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
SMF=SMF+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
SMF=SMF+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
GC TO 1C5
C** CCNTINUE
C** TEST FOR PRESENCE OF SAME-INDEX COMBINATION
C** IF (MAXSY(MP).EQ.0) GO TO 410
C** SET UP INDICES AND CCMPLE SUMS
C** I=IJS(MF)
C** II=I-1
C** SMC=SMT+PHX(I)*(PHX(JJ)+PHX(JJ))*PHX(JJ)*(PHX(JJ)+
SMC=SMT+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
SMR=SMR+PHX(I)*(PHX(JJ)+PHX(JJ))*PHX(JJ)*(PHX(JJ)+
1 SMF=SMF+PHX(I)*PHX(JJ)+PHX(JJ)*PHX(JJ)*PHX(JJ)+
C** CCMPLE BETA TERMS
C** BETA(M)=SMO
C** BETA(M)=SMT
410 BETA(M)=SMR*.5
EXT3(M)=SMR*.5

```











```

C*****
N=AN(MN)
NFS=MASS(K)
CALL BCDB(K,B,DB,D,CC)
E(1,1)=B
E(1,2)=C
E(1,3)=0
E(1,4)=C
E(2,1)=C
C1=(1.-NU)
RA=R(K)
GA=GAM(K)
C1=CMX(K)
LEX=DECMX(K)
REX=(3.*QT-0X)
GA2=GA**2
RXE=(3.*QX-0T)
C1X=0T*CX
CNLR=LAM2*D*N*D1/(2.*RA)
CCNLR=CNLR*DD/D
E(2,2)=B*D1/2.+LAM2*D*D1*REX**2/8.
E(2,3)=CNLR*REX
E(2,4)=0
E(3,1)=C
E(3,2)=E(2,3)
E(3,3)=LAM2*D*D1*(2.*RAN+(1.+NU)*GA2)
E(3,4)=LAM2
E(4,1)=C
E(4,2)=C
E(4,3)=-D
E(4,4)=C
F(1,1)=GA*B+DB
F(1,2)=(1.+NU)*B*N/(2.*RA)+CNLR*REX*RXE/4.
F(1,3)=B*(0X+NU*0T)+LAM2*D*D1*((1.+NU)*GA2*CX+RAN*RXE/2.)
F(1,4)=LAM2*0X
F(2,1)=-F(1,2)
F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*C*C1*REX/8.)*(2.*DEX-GA*(5.*0X
1-F(2,3))+LAM2*DD*C1*REX**2/8.
F(2,3)=CNLR*(2.*(1.+NU)*GA*0T-DEX+3.*GA*(CX-CT))+CCNLR*REX
F(2,4)=C
F(3,1)=-F(1,3)
F(3,2)=CNLR*(3.*GA*CX-GA*0T*(5.+2.*NU)-DEX)+CCNLR*REX
F(3,3)=-LAM2*D*D1*((1.+NU)*GA*0X*0T+GA**3)+2.*GA*RAN
1+LAM2*CC*D1*((1.+NU)*GA2+2.*RAN)
F(3,4)=LAM2*GA*(2.-NU)
F(4,1)=C*CX
00C14160
00014170
00014180
00014190
00014200
00014210
00014220
00014230
00014240
00014250
00014260
00014270
00014280
00014290
00014300
00014310
00014320
00014330
00014340
00014350
00014360
00014370
00014380
00014390
00014400
00014410
00014420
00014430
00014440
00014450
00014460
00014470
00014480
00014490
00014500
00014510
00014520
00014530
00014540
00014550
00014560
00014570
00014580
00014590
00014600
00014610
00014620
00014630

```



```

F(4,2)=0.*NU*GA
F(4,3)=-D.*NU*GA
F(4,4)=0.
G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-C1*B*RAN/2.-LAM2*D*D1*((1.+NU)*GA2*
1CX*2+RXE*2*RAN/8.)
2.MAS/DELS
G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-CNLR*2.*GA*(REX*RXE/8.
1+(1.+NU)*OTX)
G(1,3)=B*(DEX+GA*(CX-OT))+DB*(OX+NU*OT)-LAM2*C*D1*GA*RAN*(RXE/2.+(
1+NU)*OX)
G(1,4)=LAM2*D1*GA*CX
G(2,1)=-B*GAN*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.*(-1.+(1.+
1NU)*GA*CTX+GA/8.*(6.*OTX-7.*OX**2-3.*OT**2)-DEX/4.*(5.*CT-3.*CX))
2-CNLR/4.*REX*RXE
G(2,2)=-GA*F(2,2)+D1/2.*B*OTX-B*RAN-LAM2*C*D1*((1.+NU)*CT**2*RAN
1-CTX/8.*REX**2)
2.MAS/DELS
G(2,3)=-B*N*(OT+NU*OX)/RA+DNLR*(GA*DEX-2.*GA2*OX-2.*(1.+NU)*CT
1*FAN+REX*(GA2+OTX))-DNLR*REX*GA
G(2,4)=-NU*LAM2*OT*N/RA
G(3,1)=-B*GA*(OT+NU*OX)+LAM2*D*D1*(GA*(1.+NU))*(-GA*DEX+GA2*OX
1-CNLRAN+2.*OTX*OX)+RAN/2.*(GA*OX-CA*OT-3.*DEX)
2-LAM2*CC*D1*((1.+NU)*GA2*OX+RAN/2.*RXE)
G(3,2)=-B*N*(OT+NU*CX)/RA+DNLR*(2.*(1.+NU)*GA2
1*CT-OT*RAN)+GA*DEX+3.*GA2*(CT-OX)+CTX*REX)-CNLR*(2.*(1.+NU)*GA
2*CT+GA*REX)
G(3,3)=-B*(OX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+NU)*(CTX-RAN
1+2.*GA2)+2.*(GA2+CTX))-LAM2*OD*D1*RAN*(3.+NU)*GA
2.MAS/DELS
G(3,4)=-LAM2*(D1*CTX+NU*RAN)
G(4,1)=C*(DEX+NU*GA*OX)
G(4,2)=C*NU*OT/RA
G(4,3)=C*NU*RAN
G(4,4)=-1.
RETURN
ENC
SLROUTINE POLE(K,P,DEE,DST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS,ID,JD,PHIXB,
1PHITB)
C*****
C*****THIS SUBROUTINE PRINTS THE SOLUTION AT AN INITIAL *****
C*****PCLE*****
C*****INFLICIT LCGICAL*1 ($)*****
C*****REAL NU,MT,MX,MTH,MXT,MIS,KX,KT,KXT,LAM,LAM2,MASS*****
C*****DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),ZC(4,1),*****
C*****1Z2(4,1),Z3(4,1),ZDCT(4,1),IS(99,1),JS(99,1),ID(99,1),JD(99,1),*****
C*****2PHIXB(1),PHITB(1)*****
C(CMPON/IBL2/ N(99),MNINIT

```



```

CCMMCN /IBL3/ MO,M1,M2,M3
CCMMCN /IBL4/ KMAX,KL
CCMMCN /IBL5/ IBCINL,IBCFNL
CCMMCN /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
CCMMCN /IBL8/ LSTEP,ITR
CCMMCN /IBL10/ IFREQ,NTHMAX
CCMMCN /IBL12/ KMAX1,KMAX2,NCCNV
CCMMCN /IBLJ/ JUMP
CCMMCN /BL4/ ZF1M(4,4,99),ZF2M(4,4,99),
1 ZF3M(4,4,99),ZF4M(4,4,99),
ZT(99),MT(99),DT(99),GMT(95)
CCMMCN /BL5/ SOE,CSE,ALOAD
CCMMCN /BL6/ DI,S1
CCMMCN /BL7/ R(500),GAM(500),CMT(500)
CCMMCN /BL8/ PHIX(99),PHIT(99),PHI(99)
CCMMCN /BL10/ OMXI(500),PHEE,T0,T2
CCMMCN /BL11/ PHEN,I2N
CCMMCN /BL1A/ TDLI,IDEI
CCMMCN /BL12/ LAM2,LSDI8,LSCIN
CCMMCN /BL14/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
CCMMCN /BL15/ V3(99),W3(99)
1 DEL
CCMMCN /BL17/ TH(36)
CCMMCN /BL19/ DEOMX(500)
CCMMCN /BL20/ BX3(99),BXT3(99),BE3(99)
CCMMCN /BL27/ DELSQ,EXT1(99)
CCMMCN /BL31/ TKN,ELAST,CHAR,SIGC
CCMMCN /BL32/ TEEC,$DYNMC
CCMMCN /BL100/ DELSD
CCMMCN /BL101/ DELCAD
CCMMCN /BL102/ MASS(500)
CCMMCN /BL103/ TX(99),TTH(99),TXT(99),MX(55),MTH(59),MXT(59),
CCMMCN /BL110/ QS(99)
1 CCMMCN /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2
C *****
CALL BCR(K,BS,DB,CS,DD)
M1=M1-1
M2=M2-1
IF(K.EQ.KMAX) GO TO 301
CC 202 MN=1,MNMAXO
V1(MN)=L2(MN)
V1(MN)=V2(MN)
W1(MN)=W2(MN)
I3=3+(MN-1)*KMAX2
I2=I3-1
V2(MN)=Z(1,I3)
V2(MN)=Z(2,I3)
W2(MN)=Z(3,I3)

```







```

203 IF(M0.EQ.0) GO TO 206
I3=3+(M0-1)*KMAX2
I4=I3+1
CALL ILQAD(1,Z)
TX(M0)=BS*SI*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ
1 TX(M0)=TX(M0)
MTF(M0)=MX(M0)
206 IF(M2.EC.0) GO TO 205
I3=3+(M2-1)*KMAX2
I4=I3+1
TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL
TX(M2)=TX(M2)*ABZ
TX(M2)=-TX(M2)
TX(M2)=-TX(M2)
MTF(M2)=-MX(M2)
MX(M2)=-MX(M2)
IF(JUMP.EQ.1) GO TO 205
TX(M2L)=BS*D1*(2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))/DEL
TX(M2L)=TX(M2L)*ABZ
TX(M2L)=-TX(M2L)
TX(M2L)=TX(M2L)
MTF(M2L)=-MX(M2L)
MX(M2L)=MX(M2L)
RETURN
301 CCNTINUE
CC 302 MN=1,MNMAXC
LI(MN)=L2(MN)
V1(MN)=V2(MN)
W1(MN)=W2(MN)
PFI(MN)=0.
PFI(MN)=0.
PFI(MN)=0.
IK=KMAX1+(MN-1)*KMAX2
MX(MN)=Z(4,IK)*ABZ3
MTF(MN)=0.
MX(MN)=0.
CS(MN)=C.
TX(MN)=C.
TX(MN)=0.
TX(MN)=0.
303 IF(M1.EC.0) GO TO 303
CALL FNLPC(L,Z,PHIXB,PHITB)
PFI(M1)=PTEE*ABZ0
PFI(M1)=PHEE*ABZ0
IF(JUMP.EQ.1) GO TO 1002
PFI(M1L)=PHEN*ABZC
PFI(M1L)=-PHIX(M1L)

```

```

00016080
00016090
00016100
00016110
00016120
00016130
00016140
00016150
00016160
00016170
00016180
00016190
00016200
00016210
00016220
00016230
00016240
00016250
00016260
00016270
00016280
00016290
00016300
00016310
00016320
00016330
00016340
00016350
00016360
00016370
00016380
00016390
00016400
00016410
00016420
00016430
00016440
00016450
00016460
00016470
00016480
00016490
00016500
00016510
00016520
00016530
00016540
00016550

```



```

10C2 CCNTINUE
      II=KMAX+(M1-1)*KMAX2
      IF I=II+1
      IM1=II-1
      GAK=GAM(KL)
      CALL BCB(KL,BS,DB,CS,DD)
      CALL TLCD(KL,Z)
      PHIX=Z(3,IM1)*TDLI+QMXI(KL)*Z(1,II)
      PHIT=Z(3,II)/R(KL)+QMT(KL)*Z(2,II)
      PHII=((Z(2,II)-Z(2,IM1))*TCL+GAK*Z(2,II)+Z(1,II)/R(KL))/2.
      CS(M1)=-SIGO*TKN*LAM2*((2.-NU)*Z(4,II)-DS*DC2*(PHIT/R(KL)+PHI/R(KL))+PHIX)
      1*GAK+CI*MT(M1)*ALCAC-DS*DI*(-PHIX/R(KL)-GAK*(QMT(KL)+GAK*(CMXI(KL)
      2(KL))*PHII+(-Z(3,IM1)*TDLI-GAK*Z(3,II))/R(KL)+GAK*(CMXI(KL)
      3-OMT(KL))*Z(2,II)+CMT(KL))*Z(2,IM1)-Z(2,IM1))*TDLI)*.5)
      4/DEL
      IF(MO.EQ.O) GO TO 304
      TX(MO)=TO*ABZ
      TTH(MO)=TO*ABZ
      MTH(MO)=MX(MO)
      304 IF(M2.EQ.O) GO TO 305
      TX(M2)=T2*ABZ
      TTH(M2)=-T2*ABZ
      MTH(M2)=T2*ABZ
      MTH(M2)=-MX(M2)
      MXT(M2)=MX(M2)
      IF(JUMP.EQ.O) GO TC 305
      TX(M2L)=T2N*ABZ
      TTH(M2L)=-TX(M2L)
      TTH(M2L)=-TX(M2L)
      MTH(M2L)=-MX(M2L)
      MXT(M2L)=-MX(M2L)
      GC TO 305
      303 IF(MO.EQ.O) GO TO 306
      IKM=KMAX+(MO-1)*KMAX2
      IM1=IKM-1
      CALL TLCD(KMAX,Z)
      TX(MO)=BS*SI*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+CMXI(KMAX)*Z(3,IKM+1)
      1*ABZ-T(MC)*ABZ*ALCD
      J*ABZ-T(MC)*ABZ*ALCD
      TTH(MO)=TX(MO)
      MTH(MO)=MX(MO)
      306 IF(M2.EQ.O) GO TO 305
      IKM=KMAX+(M2-1)*KMAX2
      IM1=IKM-1
      TX(M2)=BS*DI*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL
      TTH(M2)=TX(M2)
      TTH(M2)=-TX(M2)
      MTH(M2)=-MX(M2)

```















```

C ** SET LP THIRD LOOP - TC COMPARE SUM WITH ALL CTRER MCDES *00018480
C ** CC 302 MMFT=1,MNMAX,JUMP *00018490
C ** IF WE SATISFY HERE, MODE EXISTS, GC TO INCREMENT -MAXS- CR *00018500
C ** -MAXSY-) *00018510
C ** IF(NTEST.EQ.N(MMFT)) GC TO 310 *00018520
C ** 2C2 CCNTINUE *00018530
C ** IF WE MAKE IT TO HERE, WE HAVE GENERATED A NEW MODE *00018540
C ** CC WE WANT ANY MORE NEW MODES *00018550
C ** IF(ICORFL.EQ.1) GC TO 301 *00018560
C ** IF(MNMAX.GE.MAXM) GC TO 301 *00018570
C ** INCREMENT -MNMAX- AND ESTABLISH NEW MODE NUMBER *00018580
C ** MNMAX=MNMAX+JUMP *00018590
C ** N(MNMAX)=NTEST *00018600
C ** IF(JUMP.GT.1) N(MNMAX-1)=-NTEST *00018610
C ** IF(MNMAX.GE.MAXM) ICORFL=1 *00018620
C ** IF MCDE WAS ADDED TO ITSELF, GO TC -MAXSY AND IJS- SECTICN *00018630
C ** 31C IF(MN.EQ.NM) GC TC 360 *00018640
C ** MAKE ENTRIES IN -LOCS-, -IS- AND -JS- *00018650
C ** LCCS=MAXS(MMFT)+1 *00018660
C ** MAXS(MMFT)=LOCS *00018670
C ** IJS(LOCS,MMFT)=MN *00018680
C ** JS(LOCS,MMFT)=MM *00018690
C ** GC TO 301 *00018700
C ** SEE IF THE SUM OF THE MCDE WITH ITSELF WAS THE O-TH MCDE *00018710
C ** 36C IF(MN.EQ.C) GC TO 301 *00018720
C ** IF HERE, IT WASN-T, MAKE ENTRIES IN -MAXSY- AND -IJS- *00018730
C ** MAXSY(MMFT)=1 *00018740
C ** IJS(MMFT)=MN *00018750
C ** CCNTINUE *00018760
C ** MNINIT=MNMAXC+JUMP *00018770
C ** IF(ICORFL.GT.0) IPASS=IPASS+1 *00018780
C ** 3C1 *00018790
C ** *00018800
C ** *00018810
C ** *00018820
C ** *00018830
C ** *00018840
C ** *00018850
C ** *00018860
C ** *00018870
C ** *00018880
C ** *00018890
C ** *00018900
C ** *00018910
C ** *00018920
C ** *00018930
C ** *00018940
C ** *00018950

```



```

IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRIX (P,X,ZC,Z2,Z3,CEE,
1CST)
RETURN
END
SUBROUTINE TEAETA(K,Z,IS,JS,ID,JC)
C*****
C THIS SUBROUTINE CALCULATES THE INPLANE FORCES AND CARRIES CUT
C THE MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE ETA
C NCN-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K. THE ARRAYS
C IS,JS,ID,JS,IJS,MAXS,MAXC,MAXSY PREFAREC IN SUBROUTINE
C MODES ARE USED HERE
C*****
C*****
REAL NU,MT
DIMENSION Z(4,1),IS(95,1),JS(99,1),ID(95,1),JC(99,1)
COMMON /IBL1/ MNMAX
COMMON /IBL2/ N(95),MNINIT
COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(59)
COMMON /IBL8/ LSTEP,ITR
COMMON /IBL13/ ITRMAX,LSMAX
5/IBLJ/ JUMP
COMMON /BL5/ TT(99),MT(99),DT(95),DMT(99)
COMMON /BL6/ SOE,CSE,ALGAD
COMMON /BL7/ D1,SI
COMMON /BL8/ R(500),GAM(500),DMT(500)
COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)
COMMON /BL11/ QMXI(500),PHEE,TO,T2
COMMON /BL12/ TDLI,TDEL
COMMON /BL15/ NU,UI(99),V1(99),V2(95),U2(55),U3(55),
1 V3(55),W3(99)
COMMON /BL27/ BX3(95),BT3(99),BXT3(55),BE3(55)
COMMON /BL28/ EXX3(99),ETI3(99),ETX3(99),EX3(99),ET3(99)
COMMON /BLPHS/ PHX(99),PHI(55)
C DIMENSION ICN TX(99),TTH(99),TXI(99)
C*****
C*****
RRA=1./R(K)
GA=GAM(K)
CX=CMXI(K)
CT=CMT(K)
CALL BCE(K,BS,DB,CS,DD)
DC 1 N=1,MNMAXO
PHIX(M)=PHIX(M)+PHX(M)
PHIT(M)=PHIT(M)+PHT(M)
EN=N(M)
CALL TLGAC(K,Z)
TTS=TT(M)*ALGAD
EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+CSE*(BX3(M)+EE3(M))
ET=EN*V2(M)*RRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M))
EXT=.5*(TDLI*(V3(M)-V1(M))-EN*U2(M)*RRA-GA*V2(M))+CSE*EXT3(M)

```



TX(M)=BS*(EX+NU*ET)-TTS	00C19440
TTH(M)=BS*(ET+NU*EX)-TTS	00019450
TX(M)=BS*CI*EXT	00019460
IF(JUMP.EQ.2) GO TC 1111	00C19470
1	00019480
1	00C19490
1	00C19500
1	00C19510
1	00C19520
1	00C19530
1	00019540
1	00C19550
1	00C19560
1	00019570
1	00C19580
1	00019590
1	00019600
1	00C19610
1	00019620
1	00C19630
1	00C19640
1	00019650
1	00C19660
1	00C19670
1	00019680
1	00019690
1	00019700
1	00019710
1	00C19720
1	00019730
1	00C19740
1	00019750
1	00019760
1	00C19770
1	00C19780
1	00019790
1	00C19800
1	00C19810
1	00019820
1	00C19830
1	00C19840
1	00C19850
1	00019860
1	00C19870
1	00019880
1	00019890
1	00019900
1	00C19910

TX(M)=BS*(EX+NU*ET)-TTS	00C19440
TTH(M)=BS*(ET+NU*EX)-TTS	00019450
TX(M)=BS*CI*EXT	00019460
IF(JUMP.EQ.2) GO TC 1111	00C19470
1	00019480
1	00C19490
1	00C19500
1	00C19510
1	00C19520
1	00C19530
1	00019540
1	00C19550
1	00C19560
1	00019570
1	00C19580
1	00019590
1	00019600
1	00C19610
1	00019620
1	00C19630
1	00C19640
1	00019650
1	00C19660
1	00C19670
1	00019680
1	00019690
1	00019700
1	00019710
1	00C19720
1	00019730
1	00C19740
1	00019750
1	00019760
1	00C19770
1	00C19780
1	00019790
1	00C19800
1	00C19810
1	00019820
1	00C19830
1	00C19840
1	00C19850
1	00019860
1	00C19870
1	00019880
1	00019890
1	00019900
1	00C19910

TX(M)=BS*(EX+NU*ET)-TTS	00C19440
TTH(M)=BS*(ET+NU*EX)-TTS	00019450
TX(M)=BS*CI*EXT	00019460
IF(JUMP.EQ.2) GO TC 1111	00C19470
1	00019480
1	00C19490
1	00C19500
1	00C19510
1	00C19520
1	00C19530
1	00019540
1	00C19550
1	00C19560
1	00019570
1	00C19580
1	00019590
1	00019600
1	00C19610
1	00019620
1	00C19630
1	00C19640
1	00019650
1	00C19660
1	00C19670
1	00019680
1	00019690
1	00019700
1	00019710
1	00C19720
1	00019730
1	00C19740
1	00019750
1	00019760
1	00C19770
1	00C19780
1	00019790
1	00C19800
1	00C19810
1	00019820
1	00C19830
1	00C19840
1	00C19850
1	00019860
1	00C19870
1	00019880
1	00019890
1	00019900
1	00C19910

TX(M)=BS*(EX+NU*ET)-TTS	00C19440
TTH(M)=BS*(ET+NU*EX)-TTS	00019450
TX(M)=BS*CI*EXT	00019460
IF(JUMP.EQ.2) GO TC 1111	00C19470
1	00019480
1	00C19490
1	00C19500
1	00C19510
1	00C19520
1	00C19530
1	00019540
1	00C19550
1	00C19560
1	00019570
1	00C19580
1	00019590
1	00019600
1	00C19610
1	00019620
1	00C19630
1	00C19640
1	00019650
1	00C19660
1	00C19670
1	00019680
1	00019690
1	00019700
1	00019710
1	00C19720
1	00019730
1	00C19740
1	00019750
1	00019760
1	00C19770
1	00C19780
1	00019790
1	00C19800
1	00C19810
1	00019820
1	00C19830
1	00C19840
1	00C19850
1	00019860
1	00C19870
1	00019880
1	00019890
1	00019900
1	00C19910















```

C** 101 MAXL=MAXS(MP)
C** TEST FOR PRESENCE OF SUMMATION COMBINATIONS
C** IF (MAXL.EQ.0) GO TO 102
C** SET UP COUPLING INDICES AND TEST FOR MODE 1
C** DC 103 L=1,MAXL
C** I=IS(L,MP)
C** J=JS(L,MP)
C** II=I-1
C** JJ=J-1
C** IF (I.EQ.1) GO TO 103
C** CCMPLE SUMS FOR ASYMMETRIC SUMMATION COMBINATIONS
C** SMF=SMF+PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)
C** 1 SMS=SMS-PHIT(I)*TTH(JJ)-PHIT(J)*TTH(II)+PHIT(II)*TTH(J)
C** 1 SMV=SMV+PHIT(I)*TTH(JJ)+PHIT(J)*TTH(II)+PHIT(II)*TTH(J)
C** 1 SME=SME+PHIX(I)*TX(JJ)+PHIX(J)*TX(II)-PHIX(II)*TX(J)
C** 1 SMN=SMN-PHIX(JJ)*TX(I)
C** 1 SMT=SMT-PHI(I)*TX(JJ)-PHI(J)*TTH(JJ)+PHI(II)*TTH(J)
C** 1 CCNTINUE
C** TEST FOR PRESENCE OF DIFFERENCE COMBINATIONS
C** 102 MAXL=MAXD(MP)
C** IF (MAXL.EQ.0) GO TO 104
C** SET UP COUPLING MODES- INDICES AND TEST FOR MCDE 1
C** DC 105 L=1,MAXL
C** I=IC(L,MP)
C** II=I-1
C** JJ=J-1
C** IF (J.EQ.1) GO TO 123
C**
C00021360
C00021370
C00021380
C00021390
C00021400
C00021410
C00021420
C00021430
C00021440
C00021450
C00021460
C00021470
C00021480
C00021490
C00021500
C00021510
C00021520
C00021530
C00021540
C00021550
C00021560
C00021570
C00021580
C00021590
C00021600
C00021610
C00021620
C00021630
C00021640
C00021650
C00021660
C00021670
C00021680
C00021690
C00021700
C00021710
C00021720
C00021730
C00021740
C00021750
C00021760
C00021770
C00021780
C00021790
C00021800
C00021810
C00021820
C00021830

```



```

C      CCMPLE SUMS FOR ASYMMETRIC DIFFERENCE COMBINATIONS
C      SMF=SMF-P*IX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)
1      SMS=SMS+PHIT(JJ)*TX(I)+PHIT(J)*TTH(II)+P*IT(II)*TTH(J)
1      SMV=SMV+PHIT(JJ)*TTH(I)+PHIT(J)*TX(II)+P*IT(II)*TX(J)
1      SME=SME+PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+P*IX(II)*TX(J)
1      SMN=SMN+PHI(I)*TX(JJ)+PHI(J)*TX(II)+PHI(II)*TX(J)
1      SMT=SMT+PHI(I)*TTH(JJ)+PHI(J)*TTH(II)+PHI(II)*TTH(J)
1      GC TO 1C5
C      EXECUTE BELOW IF J=1 IN DIFF-COMB (OR I=1 IN SUM-COMB)
C      SMF=SMF+(PHIX(I)*TX(II)+PHIX(II)*TX(I))*2.0
C      SMS=SMS+(PHIT(I)*TTH(II)+PHIT(II)*TTH(I))*2.0
C      SMV=SMV+(PHIX(I)*TX(II)+PHIX(II)*TX(I))*2.0
C      SMN=SMN+(PHI(I)*TTH(II)+PHI(II)*TTH(I))*2.0
C      SMT=SMT+(PHI(I)*TTH(I)+PHI(II)*TTH(II))*2.0
C      CONTINUE
C      TEXT FOR PRESENCE CF SAME-INDEX COMBINATION
C      1C4 IF (MAXSY(MP).EQ.0) GC TO 410
C      SET UP COUPLING MCDSES- INDICES AND COMPILE SUMS
C      I=1JS(MP)
C      I=I-1
C      SMF=SMF+P*IX(I)*TX(II)+PHIX(II)*TX(I)
C      SMS=SMS+PHIT(II)*TTH(I)+PHIT(I)*TTH(II)
C      SMV=SMV+P*IX(I)*TX(II)-PHIX(II)*TX(I)
C      SMN=SMN-P*PHI(I)*TTH(II)+PHI(II)*TTH(I)
C      SMT=SMT-P*PHI(I)*TTH(I)+PHI(II)*TTH(II)
C      CONTINUE
C      PREP+RE -AETA- TERMS
C      EXX3(M)=SMF*0.5
C      ETX3(M)=SMS*0.5
C      ETX3(M)=SMV*0.5
C      EXT3(M)=SME*0.5

```



```

EX2(M) = SMA*0.5
EI2(M) = SMT*0.5
CCCONTINUE
C**45
C**222
C**UP-CATE U,V,W AND RETURN
C**
C**DC 30 M=1,MNMAXD
C**
C**U1(M)=U2(M)
C**V1(M)=V2(M)
C**W1(M)=W2(M)
C**L2(M)=U3(M)
C**V2(M)=V3(M)
C**W2(M)=W3(M)
C**RETURN
C**
C**ENC
C**ELCCK DATA
C**CCMNCN /BL27/ BX3
C**CCMNCN /BL28/ EXX3
C**CCMNCN /BL29/ BX1
C**
C**1 CCMNCN /BL30/ BX12
C**1 CCMNCN /BL30/ EXX1
C**
C**CCMNCN /BL31/ DELSG,EX11
C**REAL BX3(99)/99*0./,BT3(99)/99*0./,ETX3(99)/99*0./,BE3(99)/99*0./,
C**1 EXX3(99)/99*0./,ET3(99)/99*0./,ETX3(99)/99*0./,EX1(99)/99*0./,
C**2 EXX1(99)/99*0./,ET1(99)/99*0./,BX1(99)/99*0./,ET1(99)/99*0./,
C**3 BX11(99)/99*0./,BE1(99)/99*0./,BX2(99)/99*0./,BT2(99)/99*0./,
C**4 BX12(99)/99*0./,BE2(99)/99*0./,EXX1(99)/99*0./,ET1(99)/99*0./,
C**5 ETX1(99)/99*0./,EX1(99)/99*0./,EXX2(99)/99*0./,
C**6 ET12(99)/99*0./,ETX2(99)/99*0./,EX12(99)/99*0./,EX2(99)/99*0./,
C**7 ET12(99)/99*0./,DELSC/1*0./,EX11(99)/99*0./
C**ENC
C**//GC.SYSIN DC *

```

DATA CAFCS GO HERE



## APPENDIX B

### LISTING OF OUTPUT FROM EXAMPLE PROBLEM



1.500 TEST CASE, INDICATIVELY TYPED CASE

--INPUT DATA RECORD--

THE BOUNDARY CONDITIONS ARE:

AT THE INITIAL EDGE

--OMEGA BAR--				--LAMBDA BAR--				--E--			
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0
AT THE FINAL EDGE											

--OMEGA BAR--				--LAMBDA BAR--				--E--			
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0
( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0	( 0.0	0.0	0.0	0.0

NUMBER OF STATIONS----- 31  
 NUMBER OF MODES----- 4  
 INCREMENTAL TIME----- 1.0E-01  
 MAXIMUM NUMBER OF TIME STEPS----- 750  
 MAXIMUM NUMBER OF ITERATIONS----- 20  
 CONVERGENCE CRITERION----- 0.0100

CHARACTERISTIC SHELL DIMENSION----- 0.1500E 02  
 REFERENCE THICKNESS----- 0.5400E 00  
 REFERENCE ELASTICITY----- 0.3500E 01  
 REFERENCE STRESS----- 0.1000E 01  
 REFERENCE TIME----- 0.1000E 01  
 POISSON'S RATIO----- 0.2000E 01

CIRCUMFERENTIAL COORDINATES FOR THE POINT RECORD, IN RADIAN MEASURE, ARE:

0.0 3.141592654

THE DATA PRINTED IS DIMENSIONAL  
 EXECUTING IN SUBROUTINE MATHS



STATION	RATIOS	CSA44A	CSA45	UPPER DUFF	BRIDGE	MASS
1	0.790E-01	0.1212E-01	0.0	0.124E-00	0.0	0.1021E-03
2	0.0020E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
3	0.0102E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
4	0.0178E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
5	0.0265E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
6	0.0340E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
7	0.0409E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
8	0.0462E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
9	0.0500E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
10	0.0534E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
11	0.0566E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
12	0.0599E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
13	0.0630E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
14	0.0659E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
15	0.0685E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
16	0.0708E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
17	0.0728E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
18	0.0744E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
19	0.0757E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
20	0.0768E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
21	0.0777E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
22	0.0784E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
23	0.0789E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
24	0.0792E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
25	0.0794E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
26	0.0795E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
27	0.0796E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
28	0.0796E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
29	0.0796E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
30	0.0796E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03
31	0.0796E-01	0.107E-01	0.0	0.124E-00	0.0	0.1021E-03

STATION	P STIFFNESS	U STIFFNESS	B PRIME	O PRIME
1	0.208163E-07	0.511471E-05	0.0	0.0
2	0.208163E-07	0.511471E-05	0.0	0.0
3	0.208163E-07	0.511471E-05	0.0	0.0
4	0.208163E-07	0.511471E-05	0.0	0.0
5	0.208163E-07	0.511471E-05	0.0	0.0
6	0.208163E-07	0.511471E-05	0.0	0.0
7	0.208163E-07	0.511471E-05	0.0	0.0
8	0.208163E-07	0.511471E-05	0.0	0.0
9	0.208163E-07	0.511471E-05	0.0	0.0
10	0.208163E-07	0.511471E-05	0.0	0.0
11	0.208163E-07	0.511471E-05	0.0	0.0
12	0.208163E-07	0.511471E-05	0.0	0.0
13	0.208163E-07	0.511471E-05	0.0	0.0
14	0.208163E-07	0.511471E-05	0.0	0.0
15	0.208163E-07	0.511471E-05	0.0	0.0
16	0.208163E-07	0.511471E-05	0.0	0.0
17	0.208163E-07	0.511471E-05	0.0	0.0
18	0.208163E-07	0.511471E-05	0.0	0.0
19	0.208163E-07	0.511471E-05	0.0	0.0
20	0.208163E-07	0.511471E-05	0.0	0.0
21	0.208163E-07	0.511471E-05	0.0	0.0
22	0.208163E-07	0.511471E-05	0.0	0.0
23	0.208163E-07	0.511471E-05	0.0	0.0
24	0.208163E-07	0.511471E-05	0.0	0.0
25	0.208163E-07	0.511471E-05	0.0	0.0
26	0.208163E-07	0.511471E-05	0.0	0.0
27	0.208163E-07	0.511471E-05	0.0	0.0
28	0.208163E-07	0.511471E-05	0.0	0.0
29	0.208163E-07	0.511471E-05	0.0	0.0
30	0.208163E-07	0.511471E-05	0.0	0.0
31	0.208163E-07	0.511471E-05	0.0	0.0



INITIAL CONDITIONS FOR N= 0 FOLLOW

STATION	U	V	W	S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U	V	W	S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0



STATION	U	V	W	M S
1	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DCT	V DCT	W DCT	M S DCT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.222040E 04	0.0
3	0.0	0.0	-0.222040E 04	0.0
4	0.0	0.0	-0.222040E 04	0.0
5	0.0	0.0	-0.222040E 04	0.0
6	0.0	0.0	-0.222040E 04	0.0
7	0.0	0.0	-0.222040E 04	0.0
8	0.0	0.0	-0.222040E 04	0.0
9	0.0	0.0	-0.222040E 04	0.0
10	0.0	0.0	-0.222040E 04	0.0
11	0.0	0.0	-0.222040E 04	0.0
12	0.0	0.0	-0.222040E 04	0.0
13	0.0	0.0	-0.222040E 04	0.0
14	0.0	0.0	-0.222040E 04	0.0
15	0.0	0.0	-0.222040E 04	0.0
16	0.0	0.0	-0.222040E 04	0.0
17	0.0	0.0	-0.222040E 04	0.0
18	0.0	0.0	-0.222040E 04	0.0
19	0.0	0.0	-0.222040E 04	0.0
20	0.0	0.0	-0.222040E 04	0.0
21	0.0	0.0	-0.222040E 04	0.0
22	0.0	0.0	-0.222040E 04	0.0
23	0.0	0.0	-0.222040E 04	0.0
24	0.0	0.0	-0.222040E 04	0.0
25	0.0	0.0	-0.222040E 04	0.0
26	0.0	0.0	-0.222040E 04	0.0
27	0.0	0.0	-0.222040E 04	0.0
28	0.0	0.0	-0.222040E 04	0.0
29	0.0	0.0	-0.222040E 04	0.0
30	0.0	0.0	-0.222040E 04	0.0
31	0.0	0.0	-0.222040E 04	0.0



THE INITIAL CONDITIONS FOR N= 2 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0



THE INITIAL CONDITIONS FOR W = 4 FOLLOW

STATION	U	V	W	W DOT	W S DOT
1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	W S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0



THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.0

STATION	N S	N THETA	N SURFA	O S	N S	N THETA	N SURFA
1	-0.3459E-04	-0.9571E-03	0.0	-0.2372E-04	0.2384E-03	0.5024E-02	0.0
14	0.7629E-03	0.2451E-04	0.0	0.2531E-04	0.1838E-04	0.4444E-03	0.0
27	0.3078E-04	-0.3224E-04	0.0	-0.2760E-04	-0.6155E-03	-0.1717E-03	0.0
31	0.2031E-04	0.7875E-03	0.0	0.1322E-04	-0.0756E-02	-0.2790E-02	0.0

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.016159E-01

STATION	U	V	W	PHI C	PHI THETA	PHI
1	0.0	0.0	0.2749E-09	0.1774E-03	0.0	0.0
14	-0.6017E-02	0.0	0.1312E-00	-0.3545E-01	0.0	0.0
27	-0.4079E-02	0.0	0.3027E-01	0.461E-01	0.0	0.0
31	-0.1578E-09	0.0	-0.6352E-10	-0.5000E-09	0.0	0.0

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.016159E-01

STATION	N S	N THETA	N SURFA	O S	N S	N THETA	N SURFA
1	-0.4580E-04	-0.1275E-04	-0.3960E-02	-0.1933E-04	0.1474E-04	0.4150E-03	-0.1914E-04
14	-0.6554E-04	-0.3791E-05	-0.1549E-02	-0.2551E-03	-0.0058E-03	-0.3420E-03	0.1056E-03
27	-0.3588E-04	-0.6111E-04	0.4616E-02	0.3540E-03	0.5024E-03	0.1061E-01	-0.7260E-04
31	-0.1638E-04	-0.5050E-03	0.3203E-02	0.5090E-03	0.1454E-04	0.4284E-03	0.1160E-04

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.016159E-01

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	-0.4935E-09	0.3810E-09	0.1079E-15	-0.2664E-08
14	0.5233E-02	-0.3805E-07	-0.2074E-00	0.2520E-01	0.1761E-07	-0.2673E-08
27	0.2689E-02	-0.1181E-07	-0.6500E-01	-0.3038E-01	0.4165E-08	-0.2195E-08
31	0.3275E-10	-0.4139E-15	-0.3176E-10	0.2540E-09	-0.6094E-14	0.2155E-08



## APPENDIX C

### INPUT DATA GUIDE FOR SATANS-IIA



# INPUT DATA GUIDE FOR SATANS-I, SATANS-II, AND SATANS-IIA

CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
1	1-72	18A4	TITLE	-	ENTER ANY 72 CHARACTERS
2	1-5	I5	NO	1	THE PROBLEM NUMBER, 0<N<10000.
2	6-10	L5	\$DYNAMC	F T	FOR A STATIC ANALYSIS, SET \$DYNAMC = F. FOR A DYNAMIC ANALYSIS, SET \$DYNAMC = T.
2	11-15	I5	IMODE	0 1	FOR NO MODAL OUTPUT DATA FOR MODAL OUTPUT DATA.
2	16-20	I5	NDIMEN	0 1	DIMENSIONAL OUTPUT DATA. NONDIMENSIONAL OUTPUT.
2	21-25	I5	NTHMAX	8	SUMMED SOLUTION WILL BE PRINTED AT NTHMAX MERID- IANS, 0<NTHMAX<=36.
2	26-30	I5	IFREQ	2	SOLUTION WILL BE PRINTED AT THE FIRST STATION, EVERY SUBSEQUENT IFREQ STATION AND THE LAST STATION, 0<IFREQ<=KMAX.
2	31-35	I5	IPRINT	3	EVERY IPRINT CONVERGED SOLUTION WILL BE PRINT- ED.
2	36-40	I5	IBCINL	-1 0	IF THE SHELL HAS A POLE AT THE FIRST STATION. IF THE SHELL HAS NO POLE AT THE FIRST STATION.
2	41-45	I5	IBCFNL	-1 0	IF THE SHELL HAS A POLE AT THE LAST STATION. IF THE SHELL HAS NO POLE AT THE LAST STATION.



CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
2	46-50	I5	KMAX	35	NUMBER OF MERIDIONAL STATIONS. NOTE: KMAX<201 FOR SATANS -I WITHOUT PLCTS AND KMAX<101 FOR SATANS-I WITH PLOTS OR FOR SATANS -II. SATANS-IIA IS UNLIMITED.
2	51-55	I5	MNMAX	7	NUMBER OF SERIES COEFFICIENTS USED TO DESCRIBE THE INITIAL CONDITIONS, PRESSURE AND THERMAL LOADS (AND INITIAL IMPERFECTIONS IF USING SATANS -II OR IIA). MNMAX<=MAXM.
2	56-60	I5	MAXM	7	MAX NUMBER OF HARMONICS IN THE SOLUTION, LIMITED TO 99.
2	61-65	I5	LSMAX	1 99 3000	FOR A LINEAR ANALYSIS, USE MANY LOAD STEPS FOR A NONLINEAR STATIC ANALYSIS. FOR A DYNAMIC ANALYSIS, LSMAX IS THE NUMBER OF TIME INCREMENTS, WHERE $LSMAX = T_{MAX}/\Delta T$ .
2	66-70	I5	LCHMAX	2 0	THE NUMBER OF LOAD STEP SIZE REDUCTIONS IN A STATIC ANALYSIS, RECOMMENDED RANGE = 2-4. FOR A DYNAMIC ANALYSIS.
2	71-75	I5	ITRMAX	1 30	FOR A LINEAR ANALYSIS, THE NUMBER OF ITERATIONS AT A LOAD OR TIME STEP. FOR A NONLINEAR ANALYSIS, SUGGESTED RANGE = 10-30, UP TO 50 FOR SPECIAL CASES.
2	76-80	I5	IC	0 1	INITIAL CONDITIONS. SET TO 0 FOR A STATIC ANALYSIS, OR FOR A DYNAMIC ANALYSIS WHERE THE SFEEL IS AT REST AT T=0. FOR A DYNAMIC ANALYSIS WITH INITIAL CONDITIONS.



CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
3	1-12	E12.3	NU	0.3	POISSON'S RATIO, $\nu$ .
3	12-24	E12.3	SIG0	1000.0 1.0	REFERENCE STRESS LEVEL, $\sigma_0$ . IF THE INPUT DATA IS DIMENSIONAL.
3	24-36	E12.3	ELAST	.3E8 1.0	REFERENCE MODULUS OF ELASTICITY, $E$ . IF THE INPUT DATA IS DIMENSIONAL.
3	37-48	E12.3	TKN	.4E-2 1.0	REFERENCE THICKNESS, $h_0$ . IF THE INPUT DATA IS DIMENSIONAL.
3	49-60	E12.3	CHAR	8.16 1.0	CHARACTERISTIC SHELL DIMENSION, $a$ . IF THE INPUT DATA IS DIMENSIONAL.
3	61-72	E12.3	TEEO	0.0 .996E-5	IF A STATIC ANALYSIS. REFERENCE TIME, $T_0$ .
<hr/>					
4	1-12	E12.3	DELCAD	0.2          .1823E-6	FOR A STATIC ANALYSIS, DELCAD IS THE LOAD INCRE- MENT. IT REMAINS UN- CHANGED UNTIL THE SOLU- TION FAILS TO CONVERGE IN ITERMAX ITERATIONS, WHEN IT IS REDUCED BY A FACTOR OF FIVE. A MAXIMUM OF LCHMAX SUCH REDUCTIONS WILL OCCUR. FOR A DYNAMIC ANALYSIS, DELCAD IS THE NONDIMEN- SIONAL TIME INCREMENT.
4	13-24	E12.3	EPS	0.01	THE CONVERGENCE CRITERION RECOMMENDED RANGE OF $0.01 < EPS < 0.001$ .

CARD 4A IS ONLY INCLUDED FOR A SATANS-II OR SATANS-IIA RUN.

4A	1-5	15	JUMP	1	FOR AN ANALYSIS USING SINGLE SERIES EXPANSIONS.
				2	FOR AN ANALYSIS USING DOUBLE SERIES EXPANSIONS.
4A	5-10	15	MPERFS	0	AN ANALYSIS WITHOUT IM- PERFECTIONS.
				1	AN ANALYSIS WITH IMPERFEC- TIONS. NOTE: IF JUMP=28 MPERFS MAY BE 0 OR 1. IF JUMP =1, MPERFS MUST BE 0. IF MPERFS=1, JUMP MUST BE 2.



---

CARD COLUMN FORMAT ITEM EXAMPLE MEANING

INCLUDE AS MANY CARDS 5 AS NECESSARY TO SPECIFY NTHMAX MERIDIANS. IF NTHMAX EQUALS 0, OMIT CARD 5.

5	1-72	6E12.3	10.0	A LIST OF CIRCUMFERENTIAL COORDINATES $\Theta$ , IN DEGREES AND TENTHS, WHERE THE SOLUTION PRINTOUT IS DESIRED. THE LIST MUST HAVE NTHMAX ENTRIES.
---	------	--------	------	--

---

IF IBCINL = -1, OMIT CARDS 6 THROUGH 14. IF IBCFNL = -1, OMIT CARDS 15 THROUGH 23. CARDS 6 THROUGH 23 DESCRIBE THE BOUNDARY CONDITIONS AT THE FIRST, AND THEN AT THE LAST STATION. THE BOUNDARY CONDITIONS EXIST ON THE TOTAL VARIABLES, NOT ON THE INDIVIDUAL HARMONICS. LOADINGS APPLIED THROUGH SPECIFICATION OF BOUNDARY CONDITIONS ARE TAKEN IN THE ZERO-ETH HARMONIC (N=0) ONLY, AS THE COLUMN MATRIX  $\{x\}$  IS SET TO ZERO FOR HARMONICS GREATER THAN ZERO. THE BOUNDARY CONDITIONS ARE DIMENSIONAL. THE FORMAT OF CARDS 6 THROUGH 23 IS 4E16.8.

CARD 6,15 CARD 7,16 CARD 8,17 CARD 9,18

$$\begin{bmatrix} n(1,1) \\ n(2,1) \\ n(3,1) \\ n(4,1) \end{bmatrix} \begin{bmatrix} n(1,2) \\ n(2,2) \\ n(3,2) \\ n(4,2) \end{bmatrix} \begin{bmatrix} n(1,3) \\ n(2,3) \\ n(3,3) \\ n(4,3) \end{bmatrix} \begin{bmatrix} n(1,4) \\ n(2,4) \\ n(3,4) \\ n(4,4) \end{bmatrix} \begin{bmatrix} N \\ N \\ N \\ N \end{bmatrix} +$$

$$\begin{bmatrix} \wedge(1,1) \\ \wedge(2,1) \\ \wedge(3,1) \\ \wedge(4,1) \end{bmatrix} \begin{bmatrix} \wedge(1,2) \\ \wedge(2,2) \\ \wedge(3,2) \\ \wedge(4,2) \end{bmatrix} \begin{bmatrix} \wedge(1,3) \\ \wedge(2,3) \\ \wedge(3,3) \\ \wedge(4,3) \end{bmatrix} \begin{bmatrix} \wedge(1,4) \\ \wedge(2,4) \\ \wedge(3,4) \\ \wedge(4,4) \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ M \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \end{bmatrix}$$


---

CARD 24 IS:

1. INCLUDED FOR A SATANS-I STATIC ANALYSIS.
2. INCLUDED BUT BLANK FOR A SATANS-I DYNAMIC ANALYSIS.
3. OMITTED FOR A SATANS-II ANALYSIS.
4. INCLUDED BLANK FOR DYNAMIC USED FOR STATIC SATANS-IIA ANALYSES.

---

CARD COLUMN FORMAT ITEM EXAMPLE MEANING

24	1-2	L2	\$PLOTS	F	INDICATES PLOTS ARE NOT DESIRED.
				T	INDICATES PLOTS ARE DESIRED.
24	3-4	L2	\$MODAL	F	INDICATES PLOTS ARE FOR SUMMED SOLUTIONS ONLY.
				T	INDICATES PLOTS ARE FOR MODAL SOLUTIONS ONLY.



FOR THE REMAINDER OF CARD 24 ENTRIES, 0 INDICATES THAT NO PLOTS ARE DESIRED FOR THE PARTICULAR ITEM, AND 1 INDICATES THAT THEY ARE DESIRED. ALL GRAPHS ARE PLOTTED AS THE INDICATED ITEM VERSUS THE STATION NUMBER. IF A COMPLETE PLOT IS DESIRED, INSUTE IFREQ = 1.

CARD	COLUMN	FORMAT	ITEM	EXAMPLE	MEANING
24	5- 6	I2	IRADII	1	PLCT THE RADII AS COMPUTED BY SUBROUTINE GEOM.
24	7- 8	I2	IGAMMA	1	PLCT $P'/P$ AS COMPUTED BY SUBROUTINE GEOM.
24	9-10	I2	IOMEGS	1	PLCT $\omega_s$ AS COMPUTED BY SUBROUTINE GEOM.
24	11-12	I2	IOMEGT	1	PLCT $\omega_\theta$ AS COMPUTED BY SUBROUTINE GEOM.
24	13-14	I2	IDEGMS	1	PLCT $\omega_s'$ AS COMPUTED BY SUBROUTINE GEOM.
24	15-16	I2	IBSTIF	1	PLCT THE STIFFNESS C AS COMPUTED BY SUBROUTINE BDB.
24	17-18	I2	IDSTIF	1	PLOT THE STIFFNESS D AS COMPUTED BY THE SUBROUTINE BDB.
24	19-20	I2	IBBSTF	1	PLCT THE STIFFNESS $db/ds$ AS COMPUTED BY SUBROUTINE BDB.
24	21-22	I2	IDDSTF	1	PLCT THE STIFFNESS $dd/ds$ AS COMPUTED BY SUBROUTINE BDB.
24	23-24	I2	IPR	1	PLOT THE NORMAL COMPONENT OF THE PRESSURE LOAD.
24	25-26	I2	IPS	1	PLOT THE MERIDIONAL COMPONENT OF THE PRESSURE LOAD.
24	27-28	I2	IPT	1	PLCT THE CIRCUMFERENTIAL COMPONENT OF THE PRESSURE LOAD.
24	29-30	I2	ITT	1	PLCT THE THERMAL LOAD.
24	31-32	I2	IMT	1	PLOT THE THERMAL MOMENT.
24	33-34	I2	IDTT	1	PLCT $d/ds$ OF THE THERMAL LOAD.
24	35-36	I2	IDMT	1	PLOT $d/ds$ OF THE THERMAL MOMENT.
24	37-38	I2	INS	1	PLOT THE MERIDIONAL MEMBRANE FORCE DISTRIBUTION.



CARD	COLUMN	FORMAT	ITEM	EXAMPLE	MEANING
24	39-40	I2	INTH	1	PLCT THE CIRCUMFERENTIAL MEMBRANE FORCE DISTRIBUTION.
24	41-42	I2	INSTH	1	PLCT THE MERIDIO-CIRCUMFERENTIAL MEMBRANE FORCE DISTRIBUTION.
24	43-44	I2	IQS	1	PLCT THE TRANSVERSE FORCE DISTRIBUTION.
24	45-46	I2	IMS	1	PLCT THE MERIDIONAL MOMENT DISTRIBUTION.
24	47-48	I2	IMTH	1	PLCT THE CIRCUMFERENTIAL MOMENT DISTRIBUTION.
24	49-50	I2	IMSTH	1	PLCT THE MERIDIO-CIRCUMFERENTIAL MOMENT DISTRIBUTION.
24	51-52	I2	IU	1	PLCT THE MERIDIONAL DISPLACEMENT DISTRIBUTION.
24	53-54	I2	IV	1	PLCT THE CIRCUMFERENTIAL DISPLACEMENT DISTRIBUTION.
24	55-56	I2	IW	1	PLCT THE NORMAL DISPLACEMENT DISTRIBUTION.
24	57-58	I2	IPHS	1	PLCT THE MERIDIONAL ROTATION DISTRIBUTION.
24	59-60	I2	IPHIT	1	PLCT THE CIRCUMFERENTIAL ROTATION DISTRIBUTION.
24	61-62	I2	IPHI	1	PLCT THE MERIDIO-CIRCUMFERENTIAL ROTATION DISTRIBUTION.

---

INCERT IMPERFECTION DATA HERE FOR A SATANS-II OR SATANS-IIA ANALYSIS WITH IMPERFECTIONS. INSURE FORMAT OF THE IMPERFECTION DATA IS COMPATIBLE WITH THAT SPECIFIED IN THE USER-WRITTEN SUBROUTINE IMPERF.

---

25	1-2	I2	IRNAGN	0	INDICATES THIS IS THE ONLY RUN.
				1	INDICATES ANOTHER RUN IS TO BE MADE. ADD ANOTHER COMPLETE SET OF DATA CARDS AFTER THIS CARD IS IRNAGN= 1.



APPENDIX D

LISTING OF NEW POLE ROUTINE FOR SATANS-IIA



THE FOLLOWING CARDS ARE TO BE PLACED INTO THE FCRCE SUBROUTINE

CCMMCN /IBL5/IBCINL,IBCFNL

```

C      IN FCRCE
1C      IF(K.NE.2.OR.(K.EQ.2.AND.IBCINL.GE.0)) GO TO 501
DC 502      II=1,4
          SUMX=0.
          CC 503      L=1,4
          SUMX=SUMX+DL(II,L,N)*GEE(L)
9C2      X(II,IK1)=SUMX
9C2      X(II,IK1)=SUMX
9C1      CCNTINUE
          CC 11      I=1,4

```

THE FOLLOWING CARDS ARE TO BE PLACED INTO THE PMATRIX SUBROUTINE

```

C      IN PMATRIX
          CALL EFG(2,MN)
          CALL ABC
          CALL MATINV(A,4,G1,0,DETERM,IPIVCT,INDEX,4,ISCALE)
DC 501      II=1,4
          CC 501      JJ=1,4
          CL(II,JJ,MN)=0.
          CG(II,JJ,MN)=0.
          CF(II,JJ,MN)=0.
          IF(NN.GT.1) GO TO 12
          IF(NN.GT.0) GO TO 13
9C1

```



```

NC=MN
CL(1,1,MN)=1.
CL(1,2,MN)=1.
CL(1,3,MN)=-3.
CL(1,4,MN)=-3.
CG(3,3,MN)=4.
CG(4,4,MN)=4.
CF(3,3,MN)=1.
CF(4,4,MN)=-1.
GC TO 9C2
13 IF(MN
CL(1,1,MN)=-3.
CL(1,2,MN)=1.
CL(1,3,MN)=1.
IF(N(M1).LT.0) DL(2,2,MN)=-1
CL(3,3,MN)=1.
CL(4,4,MN)=1.
CG(1,1,MN)=4.
CF(1,1,MN)=-1.
GC TO 9C2
12 M2=MN
CL(1,1,MN)=1.
CL(1,2,MN)=1.
CL(1,3,MN)=1.
CL(1,4,MN)=-3.
CG(4,4,MN)=4.
CF(4,4,MN)=-1.
CCNTINUE
9C2 CC SC3 II=1,4
CC SC3 JJ=1,4
TTF=0.
CC SC4 L=1,4
TTF=TTP+OF(II,L,MN)*A(L,JJ)
9C3 CL0(II,JJ)=TTP
CC SC5 II=1,4
CC SC5 JJ=1,4
TTF=0.
TTC=0.
CC 906 L=1,4
TTF=TTP+CL0(II,L)*C(L,JJ)
9C6 TTC=TTP+CLC(II,L)*BEE(L,JJ)
CL1(II,JJ)=DL(II,JJ,MN)-TTP
CL2(II,JJ)=DG(II,JJ,MN)-TTC
9C5 CALL MAIINV(CLL,4,G1,G,DETERM,IPIVOT,INDEX,4,ISCALE)
CC SC7 II=1,4
CC SC7 JJ=1,4
TTF=0.
TTC=0.

```



```

      SC8 L=1,4
      TTP=CL1(I1,I1,L)*CL0(L,JJ)
      TTQ=CL1(I1,I1,L)*CL2(L,JJ)
      CL(I1,JJ,MN)=-TTP
      SC7 P(I1,JJ,IJ)=TTQ
      CC TC 11
      SC M3=MN

```



APPENDIX E

LISTING OF CARDS FOR  $\bar{V}$  AND  $\bar{V}_{MAX}$



THE FOLLOWING CARDS ARE TO BE PLACED INTO THE DYNAMIC SUBROUTINE IF NEEDED

```

C      STATEMENTS FOR MAIN TO CALCULATE VBAR
185  DENCN=.125*GMXI(KMAX)*R(KMAX)**4
CC 186  N=1,MAXM
      CNLN=0.
      NM=(M-1)*KMAX2
CC 184  K=2,KL
      KT=K+1+MM
184  CNLN=CNLN+Z(3,KT)*R(K)
      CNLN=CNLN*DEL#SOE
186  VBAR(M)=CNLN/DENCN
      ITTEST=ITTEST+1
      IF(ITTEST.NE.10) GC TG 963
      ITTEST=C
      WRITE(6,183)(LSTEP,(VBAR(M),M=1,MAXM))
183  FORMAT(/5X,'VBAR AT TIME STEP ',I4,' FOR EACH MODE IS',5E16.4)
963  CC 187  N=1,MAXM
      IF(LSTEP.EQ.1) AVB(N)=0.
187  IF(ABS(VBAR(M)).GT.AVB(M)) AVB(M)=ABS(VBAR(M))

```



## LIST OF FIGURES AND TABLES

### A. FIGURES

1. Critical step-pressure load versus  $\lambda$  axisymmetric analyses (SATANS-I versus SATANS-IIA)..... 32
2. Critical step-pressure load versus  $\lambda$  axisymmetric analyses (SATANS-IIA versus all others)..... 33
3. Peak deflection versus P, axisymmetric and asymmetric cases for various values of  $\lambda$  ..... 34
4. Critical step-pressure load versus  $\lambda$  asymmetric analyses (SATANS-I versus SATANS-IIA)..... 35
5. Peak deflection versus P for the asymmetric analyses of  $\lambda = 6$  (N=0,1 and N=0,2) ..... 36
6. Peak deflection versus P for the asymmetric analyses of  $\lambda = 6$  (N=0,1,2,3,and4, only N=0,1,and2 plotted)..... 37
7. Critical step-pressure load versus  $\lambda$  asymmetric analyses (SATANS-IIA versus all others)..... 38

### B. TABLES

1. Critical pressure loads from the static axisymmetric analyses..... 39
2. Critical step-pressure loads from the axisymmetric



/

dynamic analyses.....	39
3. Critical step-pressure loads from the dynamic asymmetric analyses and critical asymmetric harmonics...	39
4. Dynamic asymmetric analyses for $\bar{v}_{MAX}$ versus P.....	40



## LIST OF REFERENCES

1. Ball, R. E. and Burt, J. A., " Dynamic buckling of shallow spherical shells", Journal of Applied Mechanics, v. Paper No. 73-APM-A, p. 411 to 416, June 1973.
2. Stilwell, W. C., and Ball, R. E., "A Digital Computer Study of the Buckling of Shallow Spherical Caps and Truncated Hemispheres," NASA CR-1998, June 1972.
3. Ball R. E., "A Program for the Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution," Computers and Structures, Vol. 2, 1972, p. 141 to 162; also "A Computer Program for the Geometrically Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution, Theory, and Users Manual," NASA CR-1987, April 1972.
4. Huang, N. C., " Unsymmetrical Buckling of Thin Shallow Spherical Shells", Journal of Applied Mechanics, v. 31, p. 447 to 457, September 1964.
5. Huang, N. C., " Axisymmetric Dynamic Snap-Through of Elastic Clamped Shallow Spherical Caps Due to Centrally Distributed Pressures", AIAA Journal, v. 7, No. 2, p. 215 to 220, Feb. 1969.
6. Stephens, W. B., and Fulton, R. E., " Axisymmetric Static and Dynamic Buckling of Spherical Caps Due to Centrally Distributed Pressures", AIAA Journal, v. 7, No. 11, p. 2120 to 2126, November 1969.
7. Lock, M. H., Okubo, S. and Whittier, J. S.,



- "Experiments on the Snapping of a Shallow Dome under a Step Pressure Load", AIAA Journal, v. 6, No. 1, p. 1320 to 1326, July 1968.
8. Stricklin, J. A., et al., " Nonlinear Dynamic Analysis of Shells of Revolution by Matrix Displacement Method", AIAA Journal, v. 9, No. 4, p. 629 to 636, April 1971.
  9. Akkas, N., " Bifercation and Snap-Through Phenomena in Asymmetric Dynamic Analysis of Shallow Spherical Shells", Computers and Structures, v. 6, p. 241 to 251, March 1976.
  10. Ryan, E. A., "A Digital Computer Study of the Buckling of Actual Imperfect Cylinders - A Modification of the Computer Program SATANS - Theory and User's Manual for SATANS- I and SATANS-II," A. E. Thesis, Naval Postgraduate School, June 1970.
  11. Ball, R. E. and Ryan, B. A., " Computer Analysis of Buckling of Imperfect Shells", Journal of the Structural Division, v. 99, No. ST10, p. 2097 to 2108, October 1973.
  12. Ball, R. E. et al., " A Comparison of Computer Results for the Dynamic Response of the LMSC Truncated Cone", Computers and Structures, v. 4, p. 485 to 498, July 1973.
  13. Aerospace Structures Information and Analysis Center,  
AFFDL/FBR,  
Wright-Patterson AFB,  
Ohio 45433



# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4. Assoc Professor R. E. Ball, Code 67Bp Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5. Professor Gilles Cantin, Code 69Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Professor Johann Arbocz Firestone Flight Science Laboratory California Institute of Technology Pasadena, CA. 91109	1
7. LT Michael D. Shutt, USN 7186 Emerald Ave. Dublin, California 94566	1



Thesis  
S49325

Shutt

168188

c.1

Static and dynamic  
buckling of shallow  
spherical shells sub-  
jected to axisymmetric  
and nearly axisym-  
metric step pressure  
loads using SATANS-IIA,  
a modified version of  
SATANS-II..

Thesis

168188

S49325 Shutt

c.1

Static and dynamic  
buckling of shallow  
spherical shells sub-  
jected to axisymmetric  
and nearly axisym-  
metric step pressure  
loads using SATANS-IIA,  
a modified version of  
SATANS-II.

thesS49325

Static and dynamic buckling of shallow s



3 2768 001 95377 1

DUDLEY KNOX LIBRARY